Linköping Studies in Science and Technology
Thesis No. 764

Single and Multiple Motion Field Estimation

Magnus Hemmendorff

INSTITUTE OF TECHNOLOGY
LINKÖPINGS UNIVERSITET

LIU-TEK-LIC-1999:22

Department of Electrical Engineering
Linköpings universitet, SE-581 83 Linköping, Sweden
http://www.isy.liu.se

Linköping April 1999

Single and Multiple Motion Field Estimation

© 1999 Magnus Hemmendorff

Department of Electrical Engineering
Linköpings universitet
SE-581 83 Linköping
Sweden

ISBN 91-7219-478-2 ISSN 0280-7971

Abstract

This thesis presents a framework for estimation of motion fields both for single and multiple layers. All the methods have in common that they generate or use constraints on the local motion. Motion constraints are represented by vectors whose directions describe one component of the local motion and whose magnitude indicate confidence.

Two novel methods for estimating these motion constraints are presented. Both methods take two images as input and apply orientation sensitive quadrature filters. One method is similar to a gradient method applied on the phase from the complex filter outputs. The other method is based on novel results using canonical correlation presented in this thesis.

Parametric models, e.g. affine or FEM, are used to estimate motion from constraints on local motion. In order to estimate smooth fields for models with many parameters, cost functions on deformations are introduced.

Motions of transparent multiple layers are estimated by implicit or explicit clustering of motion constraints into groups. General issues and difficulties in analysis of multiple motions are described. An extension of the known EM algorithm is presented together with experimental results on multiple transparent layers with affine motions. Good accuracy in estimation allows reconstruction of layers using a backprojection algorithm. As an alternative to the EM algorithm, this thesis also introduces a method based on higher order tensors.

A result with potential applicatications in a number of different research fields is the extension of canonical correlation to handle complex variables. Correlation is maximized using a novel method that can handle singular covariance matrices.

Acknowledgements

Many people have been important for this thesis and here follows an attempt to list those who have made the largest contributions.

Professor Hans Knutsson is my academic advisor. He has extremely lots of ideas and a good intuition. Knutsson has provided me with the embryos to many of the best results in this thesis.

Torbjörn Kronander PhD, president SECTRA-Imtec AB, is my industrial advisor and despite being overly busy, he has had a great impact on the pace of the progress of this project. He did together with my academic advisor invent the initial ideas for this project.

Mats Andersson PhD, has almost served as an assistant academic advisor and taken time to understand and discuss a major portion of this work to the level of detail.

All the people at the Computer Vision Laboratory and its manager, professor Gösta Granlund, have provided a friendly and stimulating research environment well above average. For example, Johan Wiklund maintains a very well working computer network. Gunnar Farnebäck's experience and LATEX design of licentiate thesis speeded up my work. Magnus Borga provided me with unpublished details from his research on canonical correlation.

SECTRA-Imtec AB has provided 50% financial support and I have spent half my time there to share the SECTRA spirit and widen my experience and knowledge. Thanks to SECTRA I have been able to bring my research output into commercial applications of medical imaging. http://www.sectra.se

Among our partners at medical centers are Asgrimur Ragnarsson Torbjörn Andersson MD at Örebro Regional Hospital. Lars Thorélius MD, Erik Hellgren MD, at Linköping University Hospital. Anders Persson MD and Göran Iwar MD, Hudiksvall Hospital.

Research partners have an increasing influence on our work and future plans. Thanks to Lars Wigström at Linköping University Hospital. Also thanks to Surgical Planning Laboratory at Harvard Medical School, in particular faculty members C-F Westin PhD and Professor Ron Kikinis MD.

Swedish National Board for Industrial and Technical Development (NUTEK) has provided 50% financial support for me and my colleague Mats Andersson. NUTEK has also provided partial support for Hans Knutsson and Johan Wiklund.

Contents

Chapter 1

Introduction

1.1 Motivation

All the research presented in this thesis is dedicated to medical image processing and diagnosis of cardiovascular disease, which is the leading killer throughout the industrial world. For example, according to U.S. Department of Health and Human Services[24], more than 950,000 Americans die of cardiovascular disease each year, accounting for more than 40% of all deaths. About 57 million Americans, nearly one fourth of the U.S. population, live with some form of cardiovascular disease.

This thesis presents algorithms for motion analysis that are primarily intended for angiography, i.e. medical images on blood vessels. Some parts of this work are already used in a commercial product that has been delivered for clinical use. Other parts need further development before they can be turned into commercial applications. So far, we are good in motion compensation for patients moving extremities[16, 15]. The future goal is to handle motions of a beating heart.

The motion estimation algorithms presented in this thesis are by no means limited to medical applications. Estimation of single motions is widely used and high accuracy is often crucial, e.g. in robotics and structure-from-motion applications. Multiple motion analysis is also an important field. Our methods for estimating transparent motions may enable robotics applications to handle moving shadows and reflections in windows. Our algorithms are also able to handle motions of occluding objects. Some modifications may improve performance though.

1.1.1 Cardiovascular Disease

A number of words related with cardiovascular disease are listed here.

Thrombosis	Formation of a blood clot that blocks a vessel. Can often be dissolved by drugs.
Embolism	A clot in one part of the body can break lose and block an artery in another part of the body.
Stenosis	Narrowing of a vessel. The blood sometimes finds a new way through smaller vessels.
Aneurysm	Swelling of a vessel. Often it looks like a balloon. Aneurysms that burst in the skull cause cerebral hemorrhage.
Perfusion	Blood flow through tissue.
Ischemia	Lack of oxygen in tissue. Often due to obstruction of arterial blood supply.
Capillaries	Vessels in tissue that are too small to be seen individually. On angiography images with contrast agents, they can sometimes be seen as a cloud.
infarct	Tissue death due to lack of oxygen.
Stroke	Damage to nerve cells in the brain due to lack of oxygen.

1.2 What is Digital Subtraction Angiography and why is Motion Compensation Needed?

Angiography is medical imaging on vasculature (angio = blood [vessel]). In the past, angiography was only done using conventional X-ray and contrast agents. Today it is also widely accepted to use CT[1] and there is a rapid progress in MR[2] angiography. Over the last years, more and more people seem to believe that MR is taking over a a large portion from X-ray angiography. Despite the progress and the future potential of MR, X-ray remains the gold standard, to which MR is compared, and most people seem to believe that X-ray will be indispensable even in future.

[1] Computed Tomography (CT). X-ray images are taken from different angles by a rotating X-ray source. A computer calculates a 3D reconstruction.

[2] Magnetic Resonance (MR). A combination of stationary and rotating magnetic fields are applied on the patient. These make nuclei in the atoms spin in coherence. The echoes of the rotating field can be measured. MR equipments are expensive but the total cost of using MR is not always higher than for X-ray.

1.2.1 X-ray Angiography

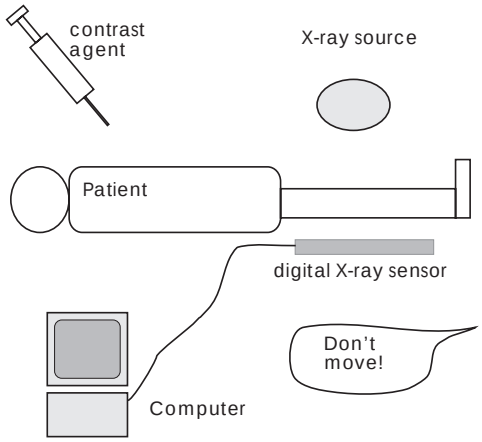

Figure 1.1: *A number of images are taken during contrast injection. The patient is told not to move, but that might be difficult.*

Figure 1.2: *angiography sequence of a leg (excerpt)*

The image sensor is usually an image intensifier tube with a CCD element at the output screen. Electronic sensors without intensifier tubes are coming. There are also image plates that are scanned by lasers and yield better image quality, but they cannot be used to acquire a sequence of images.

An ordinary dose of contrast agent is 30ml. It is injected by a long catheter directly into a vessel, upstreams of the region to be examined.

An ordinary frame rate in DSA is between 2 and 6 images per second. The frame rate is often higher in the beginning of a sequence and is decreased when the contrast agent reaches the smaller and slower vessels. Diagnosis on the heart (angiocardiography) requires a much higher frame rate.

Since blood cannot be distinguished from tissue in an X-ray image, a contrast agent is injected into an artery upstreams of the region of interest. The injection is made using a catheter, i.e. a hose that is usually inserted through arteries in the groin. Iodine-based contrast agents have significantly higher X-ray attenuation than human tissue. This means that more of the X-rays are being absorbed and fewer X-ray photons reach the sensor. The use of contrast agent enables medicals to see the vessels. By taking multiple images, during injection, it is also possible to see how the contrast agent propagates.
Unfortunately, it is often difficult to distinguish small vessels from other structure in the image. Despite the contrast agent, the images are usually dominated by bones, lungs and slowly varying thickness of the patient. The help to this problem is *image subtraction*.

1.2.2 Image Subtraction

When subtracting pixel values of two images, one taken before injection and the other taken after injection, only the vessels with contrast agent remains. Image subtraction is a simple, easy-to-understand and widely accepted method. In *digital subtraction angiography* (DSA), a reference image is taken before contrast is injected or reaches the region of interest. That reference image is then subtracted from all the images acquired after contrast injection.

Image subtraction is often a very good method. After image subtraction, nothing remains in the image, except for the contrast agent. In addition, image subtraction is a safe method and the risk of wrong diagnosis due to image subtraction is very small. Radiologists often have long experience and amazing skills in interpreting subtraction angiographies.

The predecessor of DSA, is subtraction angiography with photographic film. One film is positive and the other is negative.

1.2.3 Motions

Image subtraction requires, that nothing has moved between the images were acquired. No patient motions are allowed during image acquisition. Not surprisingly,

this makes DSA almost impossible on heart, intestines and other organs that keep moving all the time. More surprising is that motions cause problems even when the arms and legs are examined. When contrast is injected, the patient often feels a burning sensation, and move a little. Even if patients are fixated, they still move a little.

1.2.4 Pixel Shift by Hand

In conventional implementations of DSA, it is possible to compensate for motions by shifting the entire image a certain number (or fractions) of pixels. This process, called *pixel shift*, must be done manually by a medical. To save time, images with motions are often thrown away, rather than being shifted.

Except for the time required, the quality is often poor. Pixel shifts can only compensate motions that are uniform over the image, but the motions often vary over the image. This means that pixel shifts cannot achieve good quality over the entire image simultaneously.

1.2.5 Automatic Motion Compensation

We have developed automatic motion compensation[16, 15] that is a substitute to manual pixel shift. The automatic motion compensation even works for images with rotations and deformations in the image plane. Our motion compensation is very accurate for ordinary motions, including rotations and deformations. It does not matter if the motions are irregular over time. The algorithm is implemented on a dual processor Pentium-II workstation, where 1 second processing time yields enough accuracy for most images of size 512x512. A whole sequence of images can be processed without user interaction.

At the time of writing, we have attended an oral presentation of another project that addresses the same problem but with different algorithms. Their article[32] is not yet available though.

1.2.6 Objective of our Research

Our research in the past is justified by the motion compensation for angiography, and the future goal is better angiography of a beating heart. Tracking the motions of the heart in 2-dimensional X-ray images is a very difficult task. Probably, we will see several generations of motion estimation algorithms before performance is good enough. For that reason, the focus of this thesis is to solve simpler problems of multiple motions. We don't claim that the algorithms for multiple motions work on real X-ray cardio images, but we hope that research has led us closer to the solution of our specific problem. We also hope it is a step towards better analysis of multiple motions in general.

Some More Facts

Iodine-based contrast agents are no longer ionic. Ring structure molecules are popular. Despite the development of better contrast agents, some patients still have allergic reactions and chronic kidney damage. A large portion of the patients have diabetes and thus extra sensitive kidneys.

CO_2 is an alternative to iodine-based contrast agents. CO_2, which is almost transparent to X-rays, replaces the blood in the vessels and acts like a negative contrast. CO_2 is dissolved in the blood and expired by the lungs in a one-pass fashion. [22]

1.2.7 Cardiac Angiography

Angiography on a beating heart is different from angiography on peripheral parts of the body. Due to the fast motions, a higher frame rate of 12-24 images per second is used. Today, there is no technique of motion compensation and thus subtraction cannot be used. Often, angiocardiography is done with interventions and many image sequences are acquired. This means large doses of both X-ray and contrast agents. Typical images are shown in figure 1.3.

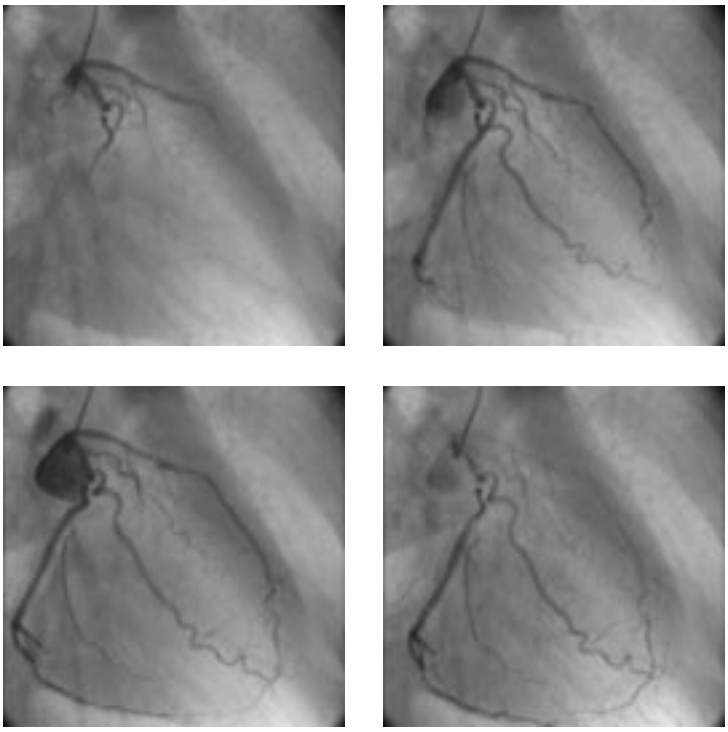

Figure 1.3: *Cardio Sequence. Frame 25, 50, 75 and 125.*

1.2.8 Interventional Angiography

A set of techniques, commonly called interventional angiography, is a cheap and simple alternative to surgery in treatment of cardiovascular disease. Thromboses and stenoses can be punctured by a wire inside the catheter. Narrowed vessels can be widened by balloons that are temporarily inserted with the catheter and inflated to high pressure. After treatment with balloons, it might be necessary to insert a tube in the vessel for it to stay open. The tubes, called *stents*, are often made of a metal grid that expands to the correct size once it has been inserted to the right position.

There are also stents that make vessels narrower, as a remedy to aneurysms or other kinds of pathological enlargement of vessels. These stents are like a hose inside the vessels. For example, sections of the aorta sometimes expand and get much too wide. A stent is fastened upstreams of the aneurysm and leads the blood past the aneurysm. The blood outside the stent coagulates and the aneurysm goes away. Other aneurysms can be treated by filling them with wire that makes the blood coagulate. Aneurysm in the brain is the leading cause of cerebral hemorrhage.

1.2.9 A Word about MR Angiography

Magnetic Resonance Angiography (MRA) has evolved rapidly over the last years. Several studies[25] indicate that MR angiography is already as good as X-ray. In addition, MR avoids problems with X-ray, such as harmful radiation. MRA can be performed without contrast agents, using velocity sensitive measurement such as phase contrast (PC) or time of flight (TOF). In practice, contrast agents may, however, be necessary in most MRA studies, but the risks are less than in X-ray. Contrast agents for MRA are less harmful and are injected intravenously, usually in the arm. This is much simpler than X-r is time consuming and requires precautions to prevent bleeding, thromboses, vessel trauma and infections.

Other advantages with MRA are abilities like 3D image acquisition. Among the disadvantages are slow image acquisition and inferior spatial resolution. Metallic implants cause image artifacts, e.g. signal void around metallic stents look like stenosis. Interventional angiography requires that all tools are non-metallic. For security, patients with pacemakers should not be exposed to magnetic resonance. Today, most people seem to believe that MRA will substitute X-ray in many situations, but to what extent is a controversial issue. Most predictions we have heard are partisan and range from “not more than today” to “almost always”.

1.3 Notations

In appendix B, there is a list of variable names in this thesis. This section is just an introduction to notations and style in this thesis.

Vectors and matrices(and tensors) are written in boldface. Matrices are upper-case and vectors are lower case. For example, boldface $\mathbf{A}$ is a matrix and boldface $\mathbf{a}$ is a vector. Vectors are always column vectors. Normal font A and a are scalars.

∇	Gradient
$\mathbf{I}$	identity matrix
$\mathbf{A}^T$	superscript T denotes transpose of matrix.
$\mathbf{A}^*$	star denotes complex conjugate and transpose of matrix.
A^*	for scalars, a star is simply a complex conjugate.
$\mathbf{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$	boldface $\mathbf{v}$ denotes image motion
$\|\mathbf{u}\|$	norm of vector $\mathbf{u}$
$\hat{\mathbf{u}} = \frac{\mathbf{u}}{\|\mathbf{u}\|}$	hat denotes normalized vector
$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}$	boldface $\mathbf{x}$ always denotes coordinate in image.

1.4 Quadrature Filters

Chapters 4 and 8 use quadrature filters that are related to Gabor filter pairs. A filter is a quadrature filter[13] if its Fourier transform, $F(\mathbf{u})$, has zero amplitude on one side of a hyperplane through the origin, i.e. there is a vector $\hat{\mathbf{n}}$ such that

$$F(\mathbf{u}) = 0 \quad \forall\, \hat{\mathbf{n}}^T \mathbf{u} \leq 0 \tag{1.1}$$

In this thesis, $\hat{\mathbf{n}}$ is called the direction of the quadrature filter. We only use quadrature filters that are real in the Fourier domain. Note that quadrature filters must be complex in the spatial domain (since $F(\mathbf{u}) \neq F^*(-\mathbf{u})$).

Quadrature filters can be optimized using a *kernel generator*, which produces efficient separable or *sequential* kernels [23, 2].

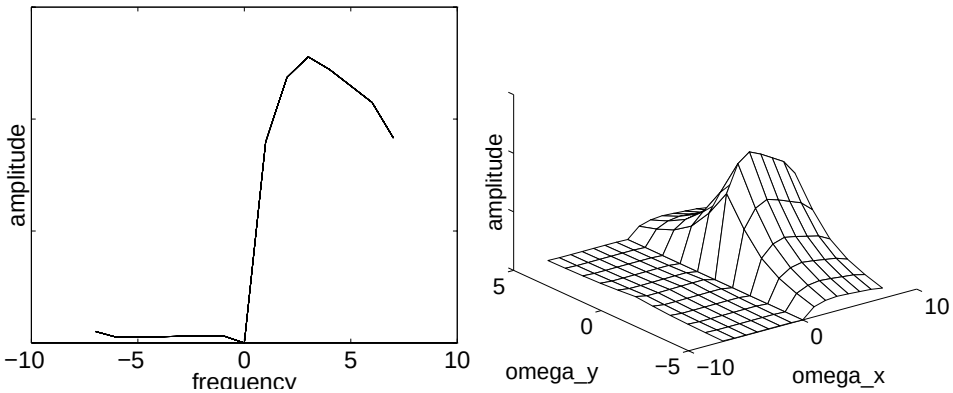

Figure 1.4: *Quadrature filters in one and two dimensions. Both filters have direction in positive x-axis.*

Chapter 2

General Issues for Single Motion Fields

This chapter is a discussion on issues in motion estimation in general. There are several existing methods, e.g. finding correlation peaks and point matching. In this thesis, the focus is methods that use two images and first estimate constraints on the local motion and then fit a motion to these.

2.1 Aperture Problem

No matter, how good tracking algorithm is used, motions cannot be unambiguously estimated in an image that only contains structure in one orientation. This is known as the *aperture problem*. For example, we can think of a moving line, viewed through a small window. Since we cannot see the line endings, it is impossible to estimate the motion component along the line. Only the orthogonal component can be estimated.

The aperture problem tells us to use big windows when estimating motions. Small windows rarely have structure in more than one orientation. How large windows depends on how far we have to go in the image, before orientation changes. In some images, e.g. figure 2.1, it might be necessary to use the entire image to estimate motion at a single coordinate.

A big window may solve the aperture problem, but fails to estimate motions locally, when motions are not uniform over the image. Chapter 3 describes how to use global motion models to overcome the aperture problem and still being able to estimate motions that are not pure translations.

2.1.1 Failure of Separable Motion Estimation Algorithms

It may seem plausible that an algorithm that estimates disparity along a scan line, can be extended to track motions in the plane. A first stupid idea was to apply

Figure 2.1: *This image contains very little structure for estimating motions in vertical direction. For sufficient accuracy, the entire image is needed. (X-ray image of a leg)*

the stereo algorithm in both horizontal and vertical direction. This would give one estimate of the motion in x-direction and another estimate in y-direction.

Although this worked pretty good in some experiments, we abandoned this approach since there is a fundamental difference between stereo algorithms and motion algorithms. The stereo algorithm assumes it can find a match along the direction of search (usually scanline). This assumption is valid for stereo images, but not for images with motions. Searching in one direction does rarely yield a correct match. Thus, we might not find a match, or even worse, find a false match. As illustrated in figure 2.2, this method is even unaware of the aperture problem.

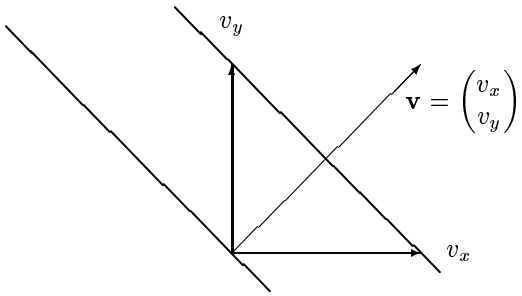

Figure 2.2: *This figure shows what happens if we track a moving line, by independently estimating the x- and y-components of the motion. The total estimate,* $\mathbf{v}$*, is seriously bad. In addition, this algorithm is unaware of the aperture problem and gives just one answer.*

2.2 Motion Constraints

Throughout this thesis, we will use constraints on local motion, like

$$c_x v_x + c_y v_y + c_t = 0 \tag{2.1}$$

where (v_x, v_y) is the local image motion and c_x, c_y and c_t are coefficients estimated locally in the image. It popular to use $c_x = dI/dx$, $c_y = dI/dy$ and $c_t = dI/dt$, where $I(\mathbf{x}, t)$ denotes the intensity of the image sequence. This method is commonly called the gradient method or optical flow[17]. A novel method of estimating motion constraints is presented in chapters 4 and 8.

If we use constraints from a single point, the motion (v_x, v_y) cannot be unambiguously determined due to the aperture problem, but by combining constraints over a larger region, the aperture problem is overcome.

For the rest of the thesis, $\mathbf{c}$ will denote a vector such that

$$\mathbf{c} = \begin{pmatrix} c_x \\ c_y \\ c_t \end{pmatrix} \tag{2.2}$$

with the property that

$$\mathbf{c}^T \begin{pmatrix} v_x \\ v_y \\ 1 \end{pmatrix} = 0. \tag{2.3}$$

Note that scaling of the constraint vector, $\mathbf{c}$, does not change the constraint on the motion, eq. 2.1. We use the magnitude of the constraint vector to denote a

confidence, i.e. a measure on how much we trust the estimate. Terminology will be sloppy and the vector **c** itself is often called *motion constraint.*

Example 2.2.1 *Intersecting Constraints: Assume motion is pure translation and we have been able to estimate motion constraints, eq. (2.1), without errors. Then there is a unique motion, v_x, v_y that satisfies all the constraints. As in figure 2.3 the motion can be solved graphically by plotting all motion constraints (v_x, v_y)-space. The intersection is the correct motion.*

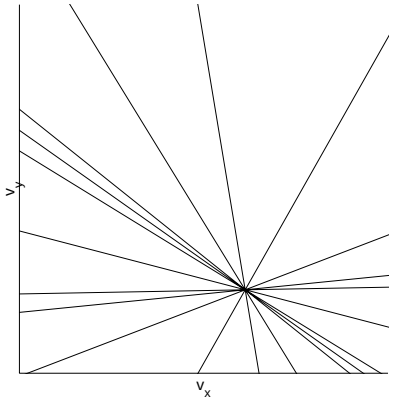

Figure 2.3: *A number of constraints. For pure translational motions, all constraints intersect at a common point in (v_x, v_y)-space.*
This representation is trivial when there is only one motion. It will be used more for better understanding of multiple motions.

For other motions than pure translations, we may use parametric motion models as described in chapter 3. We need to draw constraints in as many dimensions as there are parameters. The constraints are represented by hyperplanes that all intersect in a point corresponding to the motion.

2.3 Warping Image to Estimate Large Motions with High Accuracy

To estimate large motions with high accuracy, it is common to use a coarse to fine approach. Motions estimates from coarse scale are used to warp the image, and the estimates can be refined in finer scale. For best accuracy, more than one iteration is done in each scale. This scheme is called iterative refinement[29]. One potential problem is that a good match in coarse scale is not necessarily a good match in finer scale.

Another problem is the subpixel warp, which means resampling of the image. In imaging, unlike to audio, resampling usually means degradation since images are not perfectly bandlimited before sampling and cannot be reconstructed without error. There are several methods of interpolation, but we simply use bilinear

interpolation for maximum locality and obtain images that look good to the human eye.

Even if images warped with bilinear interpolation look quite good to the human eye, they may not look good to motion estimation algorithms. In section 2.3.3 we will present a method that avoids subpixel warps. In chapter 8 is another method presented where nothing is warped at all. Both these methods need to assume that rotations and deformations are so small that the image motion locally can be described as translation.

2.3.1 Conventional Iterative Refinement

A conventional scheme of iteratively estimating large motions with good accuracy is presented in figure 2.4. After each iteration, the original images are warped and in next iteration the error in previous iteration will be estimated. The error is supposed to converge to zero.

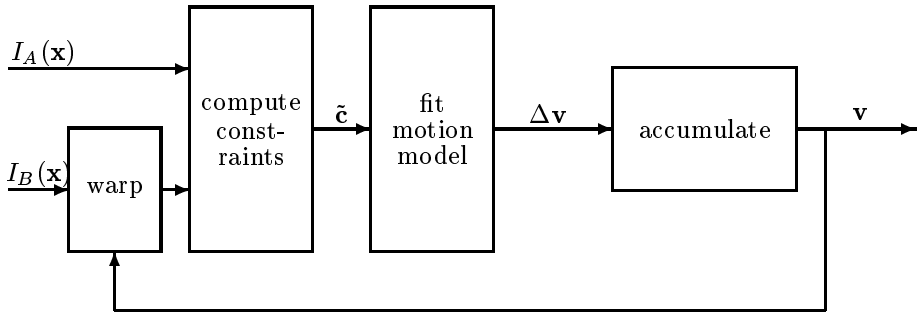

Figure 2.4: *Iterative refinement for motion estimation from two image frames, $I_A(\mathbf{x})$ and $I_B(\mathbf{x})$. Estimated motions are used to warp the image so that only a small motion remains to be estimated in the next iteration.*

2.3.2 Compensate Constraint

It turns out that the approach to warp image and estimate errors, as in figure 2.4, means difficulties when estimating multiple motions and the image is warped for each of the motion layers. The major problem is the incompatibility between constraints computed from different warps, i.e. it is complicated to use constraints computed from one warp together with constraints from another warp.

We will show how compensate for the warp directly in the constraint. Let (w_x, w_y) denote the local warp and let $(\tilde{c}_x, \tilde{c}_y, \tilde{c}_t)$ denote a motion constraint estimated from a warped image. That constraint is an estimate of the motion relative to the warp,

$$\tilde{c}_x(v_x - w_x) + \tilde{c}_y(v_y - w_y) + \tilde{c}_t = 0. \tag{2.4}$$

Thus, the correct motion constraint vector is

$$\mathbf{c} = \begin{pmatrix} \tilde{c}_x \\ \tilde{c}_y \\ \tilde{c}_t - \tilde{c}_x w_x - \tilde{c}_y w_y \end{pmatrix} \tag{2.5}$$

2.3.3 Iterative Refinement without Subpixel Warps

Thanks to eq. 2.5, we can compute the correct constraint, even if even if warp is not exact. This enables warp without subpixel accuracy where the local shifts can be rounded to integral pixels. An overview of the scheme is presented in figure 2.5. Note that unless the motion is a pure translation, it is no good to apply any spatial operations after warping with integral local shifts. In particular, we have to compute spatial gradient before warping. This limits this method to images where deformations and rotations are so small that they can locally be regarded as translations.

This method is fast since the spatial filters[1] need not be applied in each iteration. A limitation is that it cannot be used in conjunction with all possible methods of estimating motion constraints, **c**. The motion constraint must be estimated in a separable fashion, where all spatial operations are performed before operations in temporal direction. The phase-based method in chapter 4 and the conventional gradient method[17] satisfy this requirement. In the gradient method, there are no temporal operations before computing spatial derivatives and there are no spatial operations applied on the temporal derivatives.

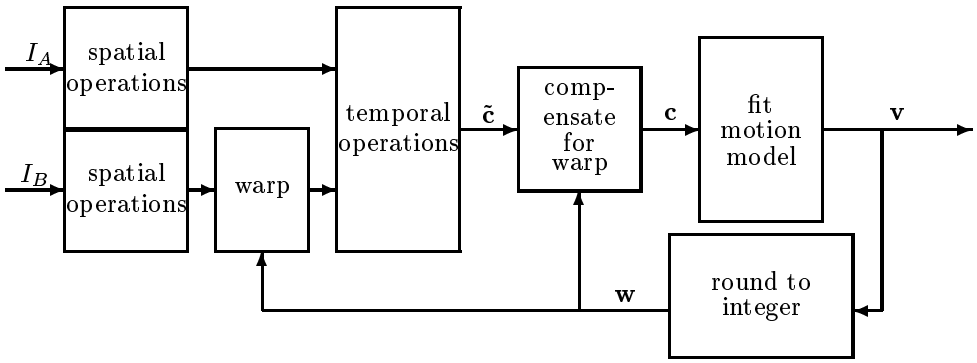

Figure 2.5: *Our scheme of warping. Instead of warping the image, a number of filter outputs are warped. Since the motion constraints,* **c** *are compensated for the warp directly, it is not necessary to warp with subpixel accuracy. (c.f figure 2.4)*

[1] We may want to use filters that are computationally expensive

Chapter 3

Parametric Motion Models

In this chapter, we assume a large number of constraints on the local motion are given (c.f. section 2.2), i.e.

$$\mathbf{c}_k^T \overline{\mathbf{v}} = 0 \quad \text{where} \quad \overline{\mathbf{v}} = \begin{pmatrix} v_x \\ v_y \\ 1 \end{pmatrix} \quad \text{and } k = 1, 2, 3, \ldots \tag{3.1}$$

Methods for computing these constraints are described in chapters 4 and 8. The focus of this chapter is how to compute the motion from these constraints, even if motion is not pure translation, i.e. the motion depends on spatial position $\mathbf{x}$

$$\mathbf{v} = \mathbf{v}(\mathbf{x}). \tag{3.2}$$

Since the constraint vectors, $\mathbf{c}$ are noisy, it does not make sense to fit a motion perfectly to these constraints. In case we would try to fit a motion field to every single constraint, the resulting estimate would be very noisy. Therefore it is necessary to fit a smooth field to the constraints. How smooth and in which way is application dependent. E.g. in orthogonal projection of a planar surface, the projection image can only be subject to translations, rotations and elongations. In case we do motion estimation on a planar surface, all estimated nonrigid deformations should be discarded. There are several different methods of fitting motions to a number of constraints. We think, in many articles[3], these methods are often associated and confused with particular methods for estimating the constraints on the local motion.

We have found it simple to use methods where the motion is represented by a number of parameters, e.g. affine motion which is described by six parameters. In this chapter, we present a general theory for parametric models where the local motion is linear with respect to the parameter vector.

3.1 Our Definition of Parametric Motion Models

A motion model describes how images move relative to one another. The motion is denoted $\mathbf{v}$ and describes how many pixels an object moves between two frames.

The motion can either be velocity or displacement. In case of two image frames $I_A(\mathbf{x})$ and $I_B(\mathbf{x})$ and no intensity variations, the image intensities are related as

$$I_A(\mathbf{x}) = I_B(\mathbf{x} + \mathbf{v}) \quad \forall \mathbf{x} \tag{3.3}$$

Unless we have pure translation, $\mathbf{v}$ is not constant over the image. Pure translation is simple, but not adequate in most applications where tracked features are being distorted or rotated. A popular motion model is the affine transformation, which can handle scaling, rotation and elongations, i.e.

$$\mathbf{v} = \begin{pmatrix} a_1 & a_2 \\ a_4 & a_5 \end{pmatrix} \mathbf{x} + \begin{pmatrix} a_3 \\ a_6 \end{pmatrix} \tag{3.4}$$

Motion models can be designed in many ways. Just for fun, let's consider one more, the quadratic motion model.

$$\mathbf{v} = \begin{pmatrix} a_7 & a_8 & a_9 \\ a_{10} & a_{11} & a_{12} \end{pmatrix} \begin{pmatrix} x^2 \\ xy \\ y^2 \end{pmatrix} + \begin{pmatrix} a_3 & a_4 \\ a_5 & a_6 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \tag{3.5}$$

We can spot a pattern. All the motion models considered so far can be written as a linear combination of basis functions. Given a set of basis functions, the motion is represented by a set of parameters, a_i. This seems to be a useful and simple way of describing almost any motion model.

$$\mathbf{v} = \sum_{i=1}^{N} a_i \, \mathbf{k}_i(\mathbf{x}) \tag{3.6}$$

To simplify notations, we arrange the coefficients in a parameter vector

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{pmatrix} \tag{3.7}$$

and the basis functions in a matrix

$$\mathbf{K}(\mathbf{x}) = \begin{pmatrix} \mathbf{k}_1(\mathbf{x}) & \mathbf{k}_2(\mathbf{x}) & \dots & \mathbf{k}_N(\mathbf{x}) \end{pmatrix} \tag{3.8}$$

and rewrite eq (3.6) as a matrix multiplication instead of a sum

$$\mathbf{v} = \mathbf{K}(\mathbf{x})\, \mathbf{a}. \tag{3.9}$$

For the rest of the thesis, boldface $\mathbf{a}$ denotes a vector of motion model parameters and $\mathbf{K}(\mathbf{x})$ is a matrix, whose columns are basis functions.

Example 3.1.1 *For the pure translation motion model,* $\mathbf{K}(\mathbf{x}) = \mathbf{I}$ *is the identity matrix and for the affine motion model*

$$\mathbf{K}(\mathbf{x}) = \begin{pmatrix} x & y & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x & y & 1 \end{pmatrix} \tag{3.10}$$

It is of course possible to swap the columns in $\mathbf{K}(\mathbf{x})$ or form new sets of basis functions.

3.1.1 Finite Element Method (FEM)

For computational efficiency in motion estimation, $\mathbf{K}(\mathbf{x})$ should be locally sparse. In other words, the basis function should have small support, i.e. $K_{ij}(\mathbf{x}) = 0$ except for in a small region in spatial domain. We might want to express the motion as a linear combination of bumps, or interpolation kernels. We have used bilinear interpolation kernels, which have small support and are continuous. Using interpolation kernels with small support is known as the finite element method. In particular, when we have bilinear interpolation kernels, solid mechanics people say we have linear elements or first order method. To get a second order method, we must have interpolation kernels with continuous derivatives.

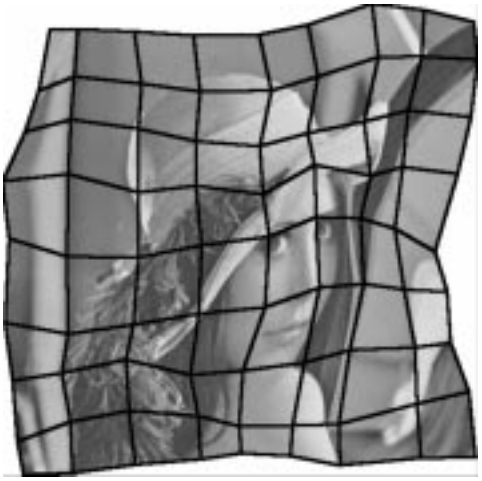

Figure 3.1: *One of our favorite motion models used to be a deformable linear mesh. The more complicated motions are, the more nodes are needed in the mesh. Each node corresponds to bilinear basis functions for horizontal and vertical motions.*

The motion model presented here can be extended to describe motion over time $\mathbf{K}(\mathbf{x}, t)$. In case motion is regular over time, a spatiotemporal model can improve accuracy. An interesting model for cyclical heart motion[35] uses truncated fourier series in temporal direction and a finite element mesh in spatial directions.

3.2 Model Based Motion Estimation

To simplify notations, the motion vector is extend with an extra entry that is always unity. For that reason, $\mathbf{K}(\mathbf{x})$ matrix and the parameter vector $\mathbf{a}$ are also extended,

$$\overline{\mathbf{v}} = \begin{pmatrix} \mathbf{v} \\ 1 \end{pmatrix}, \quad \overline{\mathbf{a}} = \begin{pmatrix} \mathbf{a} \\ 1 \end{pmatrix} \quad \text{and} \quad \overline{\mathbf{K}}(\mathbf{x}) = \begin{pmatrix} \mathbf{K}(\mathbf{x}) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}. \tag{3.11}$$

This section describes how to estimate motion model parameters from motion constraints. In other words, we have a set of motion constraint vectors, $\mathbf{c}_k$ (c.f.

section 2.2) and want to compute a the best possible parameter vector, $\mathbf{a}$ for the chosen motion model. For simplicity, we fit parameters in least square sense to constraints like $\mathbf{c}^T\overline{\mathbf{v}} = 0$ where $\overline{\mathbf{v}}(\mathbf{x}) = \overline{\mathbf{K}}(\mathbf{x})\,\overline{\mathbf{a}}$. Remember that the magnitude of the constraint vector, $\mathbf{c}$, is the confidence measure. Let $\mathbf{x}_k$ denote the spatial position of constraint $\mathbf{c}_k$ and define the following error measure that should be minimized with respect to motion model parameters

$$\begin{aligned}\varepsilon(\mathbf{a}) &= \sum_k (\mathbf{c}_k^T\,\overline{\mathbf{v}}(\mathbf{x}_k))^2 \\ &= \sum_k (\mathbf{c}_k^T\,\overline{\mathbf{K}}(\mathbf{x}_k)\,\overline{\mathbf{a}})^2 \\ &= \sum_k \overline{\mathbf{a}}^T\,\overline{\mathbf{K}}(\mathbf{x}_k)^T\,\mathbf{c}_k\,\mathbf{c}_k^T\,\overline{\mathbf{K}}(\mathbf{x}_k)\,\overline{\mathbf{a}} \\ &= \overline{\mathbf{a}}^T\,\overline{\mathbf{Q}}\,\overline{\mathbf{a}}\end{aligned} \tag{3.12}$$

where

$$\overline{\mathbf{Q}} = \sum_k \overline{\mathbf{K}}(\mathbf{x}_k)^T\,\mathbf{c}_k\,\mathbf{c}_k^T\,\overline{\mathbf{K}}(\mathbf{x}_k) \tag{3.13}$$

Since the last entry in $\overline{\mathbf{a}}$ is always one, the $\overline{\mathbf{Q}}$ matrix is splitted into a submatrix, a vector and a scalar.

$$\overline{\mathbf{Q}} = \begin{pmatrix}\mathbf{Q} & \mathbf{q} \\ \mathbf{q}^T & q\end{pmatrix}. \tag{3.14}$$

The error can be expressed as

$$\varepsilon(\mathbf{a}) = \mathbf{a}^T\,\mathbf{Q}\,\mathbf{a} + 2\mathbf{q}^T\,\mathbf{a} + q \tag{3.15}$$

and the motion model parameters are computed as

$$\mathbf{a} = \mathbf{Q}^{-1}\mathbf{q}. \tag{3.16}$$

3.3 Cost Functions

Even if motions are complicated in a global view, they may be simple locally. Motion model with many parameters allow too irregular motions. This makes the motion estimates susceptible to noise and aperture problem. The problem is even worse when using the EM algorithm in chapter 6, which gets lost in the first few iterations. It is also a problem when the basis functions in $\mathbf{K}(\mathbf{x})$ have small support, and some regions suffer from the aperture problem. Our remedy is to discourage deformations by adding a cost function to the error measure $\varepsilon(\mathbf{a})$ in eq. (3.12). For simplicity, the cost function is a quadratic form $\mathbf{a}^T\mathbf{P}\mathbf{a}$ where $\mathbf{P}$ is a symmetric matrix with nonnegative eigenvalues. Instead of minimizing that error measure, we minimize a sum of the error measure and the cost on deformations.

$$\begin{aligned}\tilde{\varepsilon}(\mathbf{a}) &= \varepsilon(\mathbf{a}) + \lambda\,\mathbf{a}^T\mathbf{P}\mathbf{a} \\ &= \mathbf{a}^T\,(\mathbf{Q} + \lambda\,\mathbf{P})\,\mathbf{a} + \mathbf{q}^T\,\mathbf{a} + q\end{aligned} \tag{3.17}$$

where $\lambda \geq 0$ is a scalar parameter that controls the stiffness, and can be included in $\mathbf{P}$, if you like. The larger lambda, the more regularization. The reason for using quadratic error measure is the computational efficiency. Compared to not using a cost function, we only have to introduce a matrix addition in eq. (3.16)

$$\mathbf{a} = (\mathbf{Q} + \lambda\,\mathbf{P})^{-1}\mathbf{Q}. \tag{3.18}$$

There is no universal way to choose λ, but in one of our implementations, it is proportional to the frobenius norm of $\mathbf{Q}$.

3.3.1 Limit on Cost

When using the EM algorithm in chapter 6, we avoid choosing λ explicitly. Instead we set a limit on the cost, i.e. we choose an upper limit on $\mathbf{a}^T\mathbf{P}\,\mathbf{a}$ and then solve for the smallest $\lambda \geq 0$ that gives a motion estimate below the limit. First we try $\lambda = 0$, and if that doesn't pass the limit, Newton-Raphson search is applied on a function that is zero at the cost limit,

$$f(\lambda) = \mathbf{a}^T(\lambda)\,\mathbf{P}\,\mathbf{a}(\lambda) - \rho_0 \tag{3.19}$$

where ρ_0 is the upper limit. Newton-Raphson solves $f(\lambda) = 0$ in a number of iterations,

$$\lambda_{n+1} = \lambda_n - \frac{f(\lambda_n)}{f'(\lambda_n)} \tag{3.20}$$

The derivative in the denominator is computed as (note that $\mathbf{a} = \mathbf{a}(\lambda)$ is a function of λ).

$$\begin{aligned} f'(\lambda) &= 2\mathbf{a}^T\mathbf{P}\,\frac{d\mathbf{a}}{d\lambda} \\ &= 2\mathbf{a}^T\mathbf{P}\,(\mathbf{Q}+\lambda\mathbf{P})^{-1}(-\mathbf{P}\mathbf{a}) \\ &= -2\mathbf{a}^T\mathbf{P}\,(\mathbf{Q}+\lambda\mathbf{P})^{-1}\mathbf{P}\mathbf{a} \end{aligned} \tag{3.21}$$

where $\frac{d\mathbf{a}}{d\lambda} = -(\mathbf{Q}+\lambda\mathbf{P})^{-1}\mathbf{P}\mathbf{a}$ was solved by differentiating $(\mathbf{Q}+\lambda\mathbf{P})\,\mathbf{a} = \mathbf{q}$, which gives that $(\mathbf{Q}+\lambda\mathbf{P})\,\frac{d\mathbf{a}}{d\lambda} + \mathbf{P}\mathbf{a} = 0$. The second derivative can be computed by differentiating again, i.e. $(\mathbf{Q}+\lambda\mathbf{P})\,\frac{d^2\mathbf{a}}{d\lambda^2} + 2\mathbf{P}\,\frac{d\mathbf{a}}{d\lambda} = 0$. This gives $\frac{d^2\mathbf{a}}{d\lambda^2} = -2(\mathbf{Q}+\lambda\mathbf{P})^{-1}\,\frac{d\mathbf{a}}{d\lambda} = 2(\mathbf{Q}+\lambda\mathbf{P})^{-1}(\mathbf{Q}+\lambda\mathbf{P})^{-1}\mathbf{P}\mathbf{a}$.

Theorem 3.3.1 $f'(\lambda) \leq 0, \quad f''(\lambda) \geq 0 \quad \forall\lambda \geq 0$
Proof: *Note that* $\mathbf{Q}$ *is symmetric, as it is defined. Without loss of generality, we can also assume* $\mathbf{P}$ *is symmetric. (Every cost function written with a non-symmetric* $\mathbf{P}$ *can also be written with a symmetric* $\mathbf{P}$*.) Then it is obvious that* $f'(\lambda) \leq 0$ *since* $(\mathbf{Q}+\lambda\mathbf{P})^{-1}$ *is positive definite.*
Next

$$\begin{aligned} f''(\lambda) &= 2\frac{d\mathbf{a}}{d\lambda}\mathbf{P}\frac{d\mathbf{a}}{d\lambda} + 2\mathbf{a}^T\mathbf{P}\frac{d^2\mathbf{a}}{d\lambda^2} \\ &= 2\,\mathbf{a}^T\mathbf{P}\,(\mathbf{Q}+\lambda\mathbf{P})^{-1}\mathbf{P}\,(\mathbf{Q}+\lambda\mathbf{P})^{-1}\mathbf{P}\mathbf{a} \\ &= 2\left((\mathbf{Q}+\lambda\mathbf{P})^{-1}\mathbf{P}\mathbf{a}\right)^T\mathbf{P}\left((\mathbf{Q}+\lambda\mathbf{P})^{-1}\mathbf{P}\mathbf{a}\right) \end{aligned} \tag{3.22}$$

Note that $f''(\lambda)$ is a quadratic form and $\mathbf{P}$ has non-negative eigenvalues, $f''(\lambda)$ cannot be negative. Thus, the sequence λ_n will decrease towards the limit, but never reach below. To get below, we modified $f(\lambda)$ by replacing ρ_0 by some value just below the limit.

3.3.2 Designing Cost Functions

There is no universal way of designing cost functions. We have tried to design cost functions without much theory. It is easy when there are only a few parameters in our motion model, but gets harder when there are more degrees of freedom. By mistake, we may forget adding a cost on deformations that should be forbidden.

We have developed a method of designing cost functions for any motion model with many parameters. We have used it to design cost functions that makes that makes a deformable mesh to locally behave like an affine transformation. The fundamental idea is to compare the estimated motions in a region, with the closest possible affine transformation.

Example 3.3.1 *To illustrate the approach of designing a cost function, let's look at an example that solves a different and much simpler problem. Assume we would measure the roughness of a signal $\mathbf{s}$. Let $\mathbf{s}_{lp}$ denote a low pass filtered version thereof. We may define roughness $= \|\mathbf{s} - \mathbf{s}_{lp}\|$. The cost is simply defined as the difference between the signal and the closest signal that is free from high frequency components. The same idea is used when designing a cost on deformations. The cost on the motion is the difference to the closest motion without non-affine deformations (locally).*

To define the cost, we locally fit an affine model to the estimated motions. The cost is the square difference between the motion estimate and the affine model that we fit to the same estimates. Let $\mathbf{K}(\tilde{\mathbf{x}})$ be the matrix including the basis functions of the affine model and let $\tilde{\mathbf{a}}$ be the affine parameters. These $\tilde{\mathbf{a}}$-parameters are locally computed from the estimated motion, $\mathbf{v}(\mathbf{x}) = \mathbf{K}(\mathbf{x})\,\mathbf{a}$.

$$\rho(\mathbf{a}) = \sum_{\text{all regions}} \min_{\tilde{\mathbf{a}}} \iint_{\text{region}} \|\mathbf{K}(\tilde{\mathbf{x}})\ \tilde{\mathbf{a}} - \mathbf{K}(\mathbf{x})\ \mathbf{a}\|^2\, dx\, dy \tag{3.23}$$

The regions must overlap, since this method does not put a cost on deformations at the borders. Evaluation of $\rho(\mathbf{a})$ into an explicit formula will give a quadratic cost function of $\mathbf{a}$ for each local region. These cost functions are summed into a global cost function that is also quadratic and can be applied as described in section 3.3. Note that the use of this method is not restricted to regularization for the finite element model. Also note that it does not have to be affine models that we locally impose. Instead of using affine motion models as a reference, we may use translational or quadratic motion models. It is even possible to use a mixture thereof, just by adding cost functions designed in different ways. For example, we can put a low cost on translations and a high cost on affine deformations.

3.4 Relation to Motion Estimation from Spatiotemporal Orientation Tensors

Knutsson and Granlund[13] have used 3D orientation tensors to estimate motion. Their tensors are a 3x3 matrix that are estimated locally in the image. To overcome the aperture problem, it may be necessary to low pass filter the tensors. Pure translation can be estimated by summing tensors over the entire image. They suggest motions should be estimated by minimizing

$$\varepsilon = \frac{\overline{\mathbf{v}}^T \mathbf{T} \overline{\mathbf{v}}}{\overline{\mathbf{v}}^T \overline{\mathbf{v}}} \tag{3.24}$$

This is similar to our least square method, described in section 3.2. In fact, our least square fit is minimization of

$$\varepsilon = \overline{\mathbf{v}}^T \mathbf{T} \overline{\mathbf{v}} \quad \text{where} \quad \mathbf{T} = \sum \mathbf{c}\mathbf{c}^T \tag{3.25}$$

Which of eq. (3.25) and eq. (3.24) gives the best motion estimate depends on how the tensors are estimated. Knutsson and Granlund use spatiotemporal filter banks to estimate the tensors. The motion is estimated without warping the image. For their tensors, one can assume that the angular error of $\begin{pmatrix} v_x & v_y & 1 \end{pmatrix}^T$ is independent of angle. For our tensors that are estimated from warped images, we assume that the absolute error of (v_x, v_y) is independent of the size of the motion.

To estimate affine motions, Farnebäck[9] expanded the tensors to size 7x7 before summing them together. His approach can be generalized to any of our motion models by replacing $\mathbf{c}_k \mathbf{c}_k^T$ by $\mathbf{T}_k$ in all eq. (3.13).

3.5 Local-Global Affine Model

In some applications the motion field is several complicated deformations. Initially, we used the finite element motion model, which models image motions like a mesh that deforms. This method become computationally expensive when the image is divided into many cells. Remind that it takes $O(N^3)$ operations to solve a linear equation system, where N denotes the number of unknowns. There are two unknowns for every node in the mesh, and the number of nodes is proportional to the square of the resolution. Thus, we have $O(N^6)$ algorithm. Another difficulty with the finite element method is to design cost function, since it must depend on the image. The cost function should depend on the distribution on magnitude of the image or motion constraint. It is also an issue how it should behave on the borders of the image. We have images where the valid region can be a circular or rectangular region of the total image. Since the valid region is not known a priori, a new cost function needs to be computed for every image.

Instead of using a global parametric model with many parameters, we use local affine models with global smoothing . The remedy to the aperture problem is low pass filtering, not cost functions. Of course, we cannot estimate motions first and

then low pass filter the motion vectors. Instead we low pass filter the coefficients of $\overline{\mathbf{Q}}$. Although the model is local, we use a global coordinate system for the affine parameters to enable low pass filtering of coefficients. The reader should convince him/herself that averaging equation system coefficients over the entire image, is equivalent to a global affine model. Averaging over a region is equivalent to using an affine model in that region. Recall that averaging is equivalent to convolution with a kernel with a constant value. You might stop up and ask what happens if we use some other non-negative kernel, e.g. a Gaussian. This is, in fact, equivalent to to weighting the constraint differently in, eq. (3.26). Usually, we are more interested in constraints in near neighborhood than far away. In order to remedy the aperture problem, it is still necessary to let motion estimates in one corner of the image influence motion estimates in opposite corner. In terms of formulas, we modify eq. (3.13) from global to local values of $\overline{\mathbf{Q}}$.

$$\overline{\mathbf{Q}}(\mathbf{x}) = \sum_k W(\mathbf{x} - \mathbf{x}_k)^2 \, \overline{\mathbf{K}}(\mathbf{x}_k)^T \, \mathbf{c}_k \, \mathbf{c}_k^T \, \overline{\mathbf{K}}(\mathbf{x}_k) \tag{3.26}$$

where $W(\mathbf{x})$ is a windowing function, e.g.

$$W(\mathbf{x}) = \frac{1}{||\mathbf{x}||^\alpha + \sigma^\alpha} \tag{3.27}$$

Where σ determines the size of the local region. A large σ will make the motion estimate more global. We recommend that the other parameter, $\alpha > 2$ (otherwise there is theoretically no locality since $\iint W(x,y)\,dx\,dy < \infty$).

Estimating motion locally rather than globally is of course more sensitive to noise and the lack of local structure. An interesting property of this method, is that if the window is strictly positive everywhere, i.e. $W(\mathbf{x}) > 0 \quad \forall \mathbf{x}$, then the local $\mathbf{Q}(\mathbf{x})$ has the same rank as the global $\mathbf{Q}$. All structure in the image contribute to all local matrices. This means that this method produces a motion estimate at every point in the image.

3.5.1 Efficient Implementation of The Local-Global Affine Model

Our implementation of the local affine motion model is almost as fast as the global affine motion model. The window function is implemented as a convolution with a low pass filter. First we compute a local version of $\overline{\mathbf{Q}}(\mathbf{x})$ using a window function that is unity in a very small neighborhood and zero everywhere else. To save computations, subsampling[1] of the matrix field is applied at the same time. This field of matrices is convolved with the window function. The window function is modified a little to be separable.

For each point in the low pass filtered matrix field, the affine parameters are solved. Since the matrix field was subsampled, the computed we applied subsampling, we need to upsample the estimated motion estimates. The upsampling is done by bilinear interpolation.

[1] We recommend that the blocksize in subsampling is significantly smaller than σ in eq. (3.27).

Efficient implementation of the low pass filter, using separable and spread kernels, makes the computational complexity reduces from $O(N^6)$ in the finite element model to $O(N^3)$.

Chapter 4

Estimation of Motion Constraints

The focus of this chapter is a novel method for estimation of constraints on the local motion, $\mathbf{c}$, as defined in section 2.2. The input is two image frames and the output is a number of (possibly conflicting) constraints for each pixel. This method can be used in conjunction with parametric motion models in chapter 3 and even for estimation of multiple motions in chapter 6.

4.1 Existing Methods

Before describing our method, we will briefly describe other existing motion estimation methods and argue why not using them.

4.1.1 Intensity Conservation Gradient Method

Traditional methods for optical flow are based on the assumption of intensity conservation over time. For X-ray images this is not valid, since (i) images in the same sequence have slightly different level due to different X-ray exposure. (ii) Contrast injection may darken the image, at least locally. (iii) Multiple layers may interfere. It may be possible to remedy these problems with prefiltering[27, 1], and advanced prefiltering can be similar to our method.

4.1.2 Point Matching

Another popular method is point matching, where a region in one image is matched to regions in another image, using some correlation scheme. Some kind of correlation measure is computed and we the algorithm chooses the match that gives maximum correlation. An alternative to maximizing correlation is to minimize a dissimilarity measure. To speed up matching, it is possible to use some gradient method or iterative search instead of explicitly computing correlation for all possible shifts.

Due to the aperture problem, point matching methods are only suitable to match regions in the image that have structure in more than one orientation, e.g. corners and line crossings. The features to track must to be found. For our medical images, point matching is not a good alternative. The amount of corners is much less than there are edges. Thus, a point matching method would only use a fraction of the information in the images.

4.1.3 Spatiotemporal Orientation Tensors

Estimating image velocity using three dimensional filter banks has proven accurate[13, 9, 10, 21] in other applications. The idea is to consider a sequence of images as a three dimensional spatiotemporal volume, of variables x, y, t. This three dimensional thinking is the same as for the gradient method[17] of optical flow, but instead of computing gradients, a set of filters are used to measure local orientation, which is the three dimensional motion vector.

The most successful method is probably[13, 21] based on a set of nine quadrature filters. The energies of each of the quadrature filter outputs are computed and combined to an orientation tensor. Except for the high accuracy, the method is good in using all the information of both edges and corners. All the information is implicit in the tensors. The aperture problem is simply overcome by low pass filtering of tensors. All information, including certainty, can be extracted using eigenvector decomposition.

Unfortunately, spatiotemporal filtering approaches are not useful in our applications. The frame rate is too low and patient motions are too large and irregular over time. In terms of signal processing, we have severe aliasing due to low temporal sampling rate. Thinking of the image sequence as a spatiotemporal volume is not helpful. Another reason for not using spatiotemporal filtering is that we want to estimate the displacement, not the velocity. In case we would use spatiotemporal filtering, we would get velocity vectors that had to be followed over time, which would result in accumulation of errors.

4.2 Phase Based Quadrature Filter Method

Using quadrature filters phase is a relatively common approach in stereo algorithms[33, 12]. The idea of using phase for motion estimation has previously been investigated by some researchers [8, 6, 11], but to our knowledge, nobody has tried this approach, which extends the accurate stereo algorithms to estimate relative motions from two image frames. Our method is almost a gradient-based method with nonlinear preprocessing of the images. To improve accuracy, a confidence measure has been added. The method presented in this thesis has been published both as an independent method[14] and in the context of angiography application[15].

Definition 4.2.1 *A filter is a quadrature filter[13] if its Fourier transform, $F(\mathbf{u})$, is zero on one side of a hyperplane through the origin, i.e. there is a direction $\hat{\mathbf{n}}$ such that*

$$F(\mathbf{u}) = 0 \quad \forall\, \hat{\mathbf{n}}^T \mathbf{u} \leq 0 \tag{4.1}$$

Quadrature filter outputs are closely related to analytic signals. Note that quadrature filters must be complex in the spatial domain. We only use filters that are real in the Fourier domain.

4.2.1 Motion Constraint Estimation

The input to the algorithm is two image frames, denoted $I_A(\mathbf{x})$ and $I_B(\mathbf{x})$ and the output is a number of motion constraints, $\mathbf{c}$, at each pixel. A number of quadrature filters are applied in parallel on each of the two image frames, producing the same number of filter outputs. The quadrature filters are tuned in different directions and frequency bands to split dissimilar features into different filter outputs, so that they do not interfere in the motion estimation. The quadrature filters also suppress undesired features like DC value and high frequencies. Unlike the conventional gradient method, our method is not sensitive to low pass variations in image intensity, that are frequent in medical X-ray images, or real world images where shadows and illumination vary.

The quadrature filters can be chosen to have different directions and different frequency bands, but all of our implementations have four filters in the same frequency band but in different directions, as shown in figure 4.1. These filters are denoted $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, $f_3(\mathbf{x})$ and $f_4(\mathbf{x})$ and are tuned in 0, 45, 90 and 135 degrees. Both the input images are convolved with each of the filters,

$$q_{A,j}(\mathbf{x}) = (f_j * I_A)(\mathbf{x}) \quad \text{and} \quad q_{B,j}(\mathbf{x}) = (f_j * I_B)(\mathbf{x}) \tag{4.2}$$

where $f_j(\mathbf{x})$ is a quadrature filter and $I_A(\mathbf{x})$ and $I_B(\mathbf{x})$ are image intensities of the two frames respectively. The phase is defined as the phase angle of the complex numbers

$$\theta_{A,j}(\mathbf{x}) = \arg q_{A,j}(\mathbf{x}) \quad \text{and} \quad \theta_{B,j}(\mathbf{x}) = \arg q_{B,j}(\mathbf{x}). \tag{4.3}$$

In all ensuing computations, we must remember that phase is always modulo 2π, but for readability we drop this in our formulas and notations. In most image points, the filter outputs are strongly dominated by one frequency, which makes the phase nearly linear in a local neighborhood. When the phase is linear, it can be represented by its value and gradient. Thus, a gradient method applied on the phase will be very accurate. Of course, the phase is not always linear in a local neighborhood, but that can be detected, and reflected by a confidence measure.

For each point in the image, and for each quadrature filter output, a constraint on the local motion is computed. To simplify notations, we drop the index, j, of the quadrature filter.

$$\mathbf{c} = \begin{pmatrix} c_x \\ c_y \\ c_t \end{pmatrix} = C \begin{pmatrix} \frac{1}{2}\frac{\partial}{\partial x}(\theta_B + \theta_A) \\ \frac{1}{2}\frac{\partial}{\partial y}(\theta_B + \theta_A) \\ \theta_B - \theta_A \end{pmatrix} \tag{4.4}$$

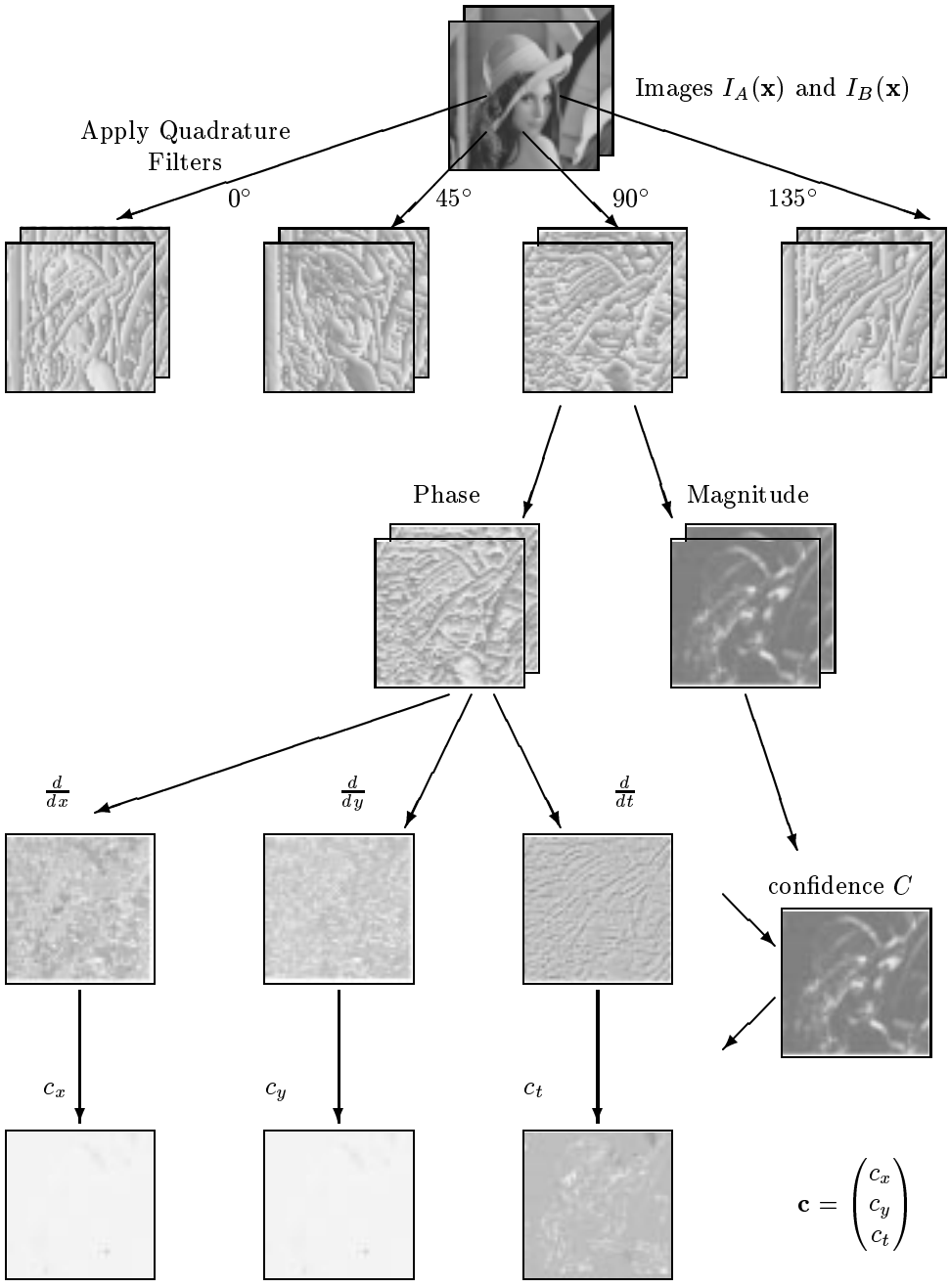

Figure 4.1: *From image to motion constraint for one direction of quadrature filters. The quadrature filter outputs are complex values, but that would take colors to illustrate so only phase images are shown. Note that phase is wrapped module* 2π.

Since the phase is locally almost linear, the derivatives can be computed as a difference between two pixels. The motion constraint vector is the spatiotemporal gradient of the phase, weighted by the confidence measure, C, which will be introduced in next section.

4.2.2 Confidence Measure

Using a confidence measure is necessary to give strong features precedence over weaker features and noise. In addition, it is necessary to avoid phase singularities[33, 20] which occur when two frequencies interfere in the filter output. These singularities must be discovered and treated as outliers. All this is done by assigning a confidence value to each constraint. Our confidence measure is inspired by the stereo disparity algorithm by Westelius [33], which in turn is inspired by [7]. It is a product of several factors, where the most important feature is the magnitude.

Our confidence measure for magnitude may seem complicated at first glance. Except for suppressing weak features, it is also sensitive to difference between the two frames. This reduces the influence of structure that only exist in one of the images, such as moving shadows, appearing objects and other features not moving according to the motion we estimate.

$$C_{mag} = \frac{|q_A|^2 |q_B|^2}{(|q_A|^2 + |q_B|^2)^{3/2}} \tag{4.5}$$

Other factors have been added to reflect whether the gradient, is sound for the specific quadrature filter in use. Negative frequencies are illegal and indicate phase singularities[20, 33].

$$C_{freq>0} = \begin{cases} 1 & \text{if } \hat{\mathbf{n}}^T \nabla\theta > 0 \ , \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

Our confidence measure is also sensitive to high frequencies, which may indicate an error in the filter output or signal probability of negative frequencies wrapping around modulo 2π.

$$C_{freq.wrap} = \begin{cases} 1 & \text{if } \|\nabla\theta\| < \omega_{max.diff}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.7}$$

where ω_{max} is related to the upper cutoff frequency of the quadrature filter. Frequencies above this are probably false and there is also an increased probability of wrap-around from negative frequencies. It might be better with a continuous drop off in confidence, but this binary function is computationally efficient since a 'C=0' can be represented by "NaN" in floating point arithmetics. We also guard for phase difference wrap arounds.

$$C_{phase-wrap} = \begin{cases} 1 & \text{if } \|\theta_B - \theta_A\| < \theta_{max}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.8}$$

When computing the frequency, it is also useful to check consistency between two images in order to avoid features that only exist in one of the images.

$$C_{freq.cons} = max\left(0,\ \frac{\omega_{max.diff} - \|\nabla\theta_A - \nabla\theta_B\|^2}{\|\nabla\theta_A\|^2 + \|\nabla\theta_B\|^2}\right) \tag{4.9}$$

where we have heuristically set $\omega_{max.diff} = 1$

Finally, the total confidence is computed as a product of all the confidence measures, i.e.

$$C = C_{freq>0} C_{freq.wrap} C_{phase-wrap} C_{mag} C_{freq.cons} \tag{4.10}$$

4.2.3 Multiple Scales and Iterative Refinement

To estimate large motions with best possible accuracy, we apply motion estimation iteratively in multiple scales. We begin at the coarsest scale in a low pass pyramid to compute a rough estimate. Then we warp the image, or its filter outputs, and do a new iteration at a finer scale. For best accuracy, we can do multiple iterations at each scale.

When estimating a motion constraint from a warped image, we get a constraint on the motion relative to the warp. Similarly, subsampling alters the estimated motion constraints to yield smaller motion estimates. It is, however, simple to compensate for the warp and subsampling. Assume the image is warped (w_x, w_y) pixels and subsampled λ octaves prior to estimation of a motion constraint, $\tilde{\mathbf{c}} = (\tilde{c}_x, \tilde{c}_y, \tilde{c}_t)$. Then we have in fact estimated that

$$\tilde{c}_x \frac{v_x - w_x}{2^\lambda} + \tilde{c}_y \frac{v_y - w_y}{2^\lambda} + \tilde{c}_t = 0. \tag{4.11}$$

Thus, the correct motion constraint is

$$\mathbf{c} = \begin{pmatrix} \tilde{c}_x \\ \tilde{c}_y \\ 2^\lambda \tilde{c}_t - \tilde{c}_x w_x - \tilde{c}_y w_y \end{pmatrix}. \tag{4.12}$$

In order to avoid subpixel warps, the method in Figure 2.5 is used.

4.3 Experimental Results

We have used the phase-based method on various image data, and it has always turned out advantageous to the conventional gradient method. One important application is motion compensation in sequences of medical X-ray images, digital subtraction angiography. The conventional gradient method fail to estimate motions accurately, due to different DC level in the frames and motions of the injected contrast agent. Suppressing low frequencies helps a lot, but still our phase-based method is superior.

4.3.1 X-ray Angiography Images

Figures 4.2 - 4.5 show a comparison for a medical X-ray angiography sequence. Image subtraction is used to extract the vessels and take away the bones and tissue. We get much less motion artifacts when using phase-based motion estimation. Constraints over the image are integrated, to fit a local-global deformable motion model[16] in least square sense. We have used four quadrature filters in different directions in conjunction with multiple scales and iterative refinement.

4.3.2 Synthetic Images

We have also compared accuracy on images where motions come from synthetic shifts. A real world test image has been shifted different amounts in different directions. To avoid influence from subpixel warps, the image has been subsampled after the warp. One might expect the conventional gradient method works pretty good on these images that have perfect intensity conservation between frames. But still, our phase-based method is more accurate, as shown in figure 4.6.

4.3.3 Synthetic Images with Disturbance

In angiography, the contrast injection causes disturbing changes in the image, that may also disturb the motion estimation. We have made an experiment on synthetic images to show that our phase-based motion estimation is less susceptible to such disturbance than the the conventional gradient method, often referred as optical flow[17]. We have used synthetically shifted images to evaluate the accuracy when one of the image frames is disturbed. We have used a popular reference image, Lena256x256, which has been shifted and then subsampled to hide artifacts due to subpixel shifts. The shifts are in all possible directions and we have computed the average performance for all shifts of the same distance.

As shown in figure 4.7, our novel method performs significantly better. Since we use iterative refinement, it is most relevant to study performance for shifts less than $\sqrt{2}/2 \approx 0.7$ pixels. After convergence, the warp reduces motion to something less than a half pixel in each of x and y directions.

4.4 Future Development

The confidence measure in this thesis is designed without much theory and experiments. It might be possible to get better accuracy with application specific confidence measures. For instance, in some applications it may be more or less important to check consistency between frames. In general, it can be that the confidence measure factor on magnitude, eq. (4.5) should depend on the noise level. Instead of being linear to magnitude, it should be a sigmoid function that give almost equal confidence to all features that are well above the noise level.

Figure 4.2: *Original X-ray images*

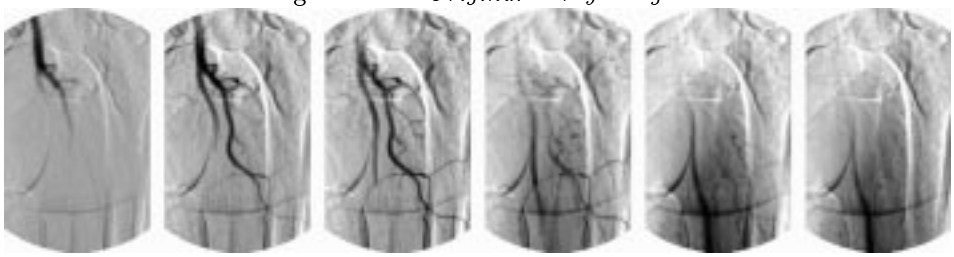

Figure 4.3: *Subtraction Images, no motion compensation*

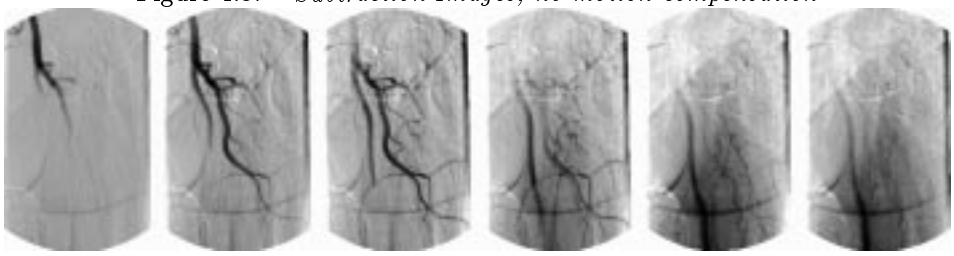

Figure 4.4: *Subtraction Images, motion compensation based on conventional gradient method, after filtering out low frequencies.*

Figure 4.5: *Subtraction Images, motion compensation based on our phase-based method. Note there are less artifacts compared to figure 4.4 (the confidence measure differs[16] slightly from the text.)*

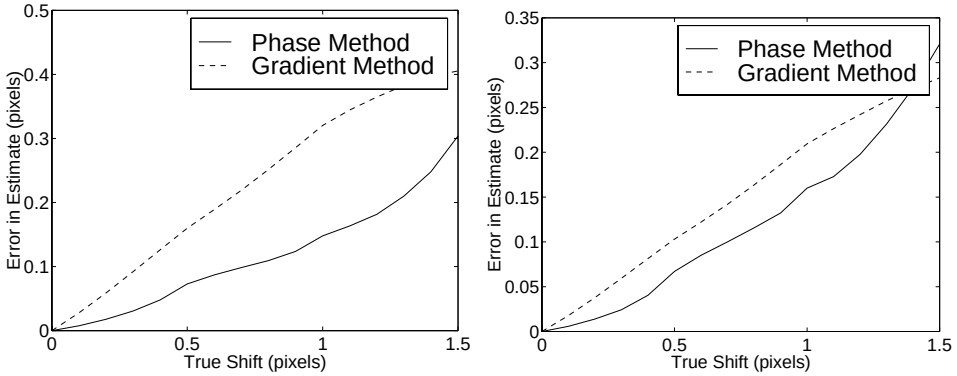

Figure 4.6: *The phase-based method is more accurate than the conventional gradient method. These plots show a comparison on images(Lena 256x256 and Debbie 128x128) that are shifted synthetically. One pass estimation – no iterative refinement.*

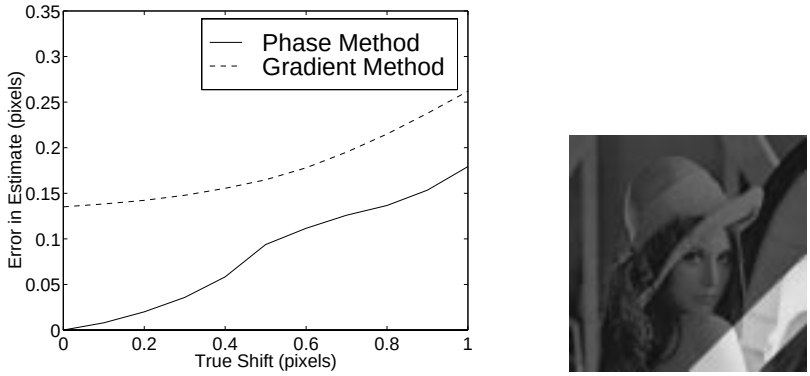

Figure 4.7: The phase-based method is more accurate than the conventional gradient method. This figure shows a comparison on images(Lena 256x256) that are shifted synthetically(shifting Lena512x512 before subsampling). One of the image frames has been disturbed by adding a transparent stripe across the image, in order to simulate a contrast bolus. (One pass estimation, i.e. no iterative refinement).

Chapter 5

General Problems in Multiple Motion Analysis

5.1 Introduction

In estimation of multiple motions, there is a number of difficulties that are not present in estimation of single motion. In the general case, estimation of multiple motions is a very difficult problem. All motions in the image need to be classified and clustered into an unknown number of fields described by unknown models. Even counting the number of motions in an image is a problem. This requires some criteria to tell how different two motions must be before they are classified as two motions instead of one. In the algorithms presented in chapter 6, it is assumed that the number of motions are known a priori.

Multiple motion problems can be classified into transparent or occluding motions. The case of occluding motions is the most common in real world images, e.g. a scene of multiple opaque objects at different depth, moving at different image velocity. The focus our research is, however, primarily dedicated to the problem of transparent motions. In X-ray images, we get a projection of structure at different depth. The X-rays goes through all parts and nothing is occluded. The logarithm of the image is thus the sum of all X-ray attenuation at all different depths. Our approximation model of the human body is a set of transparent layers that move independently. For example, an approximate model of X-ray images on the heart might be four transparent layers. Two layers are the front and back ribs and the other two layers are the front and back wall of the heart.

5.2 Motion Constraints

In the estimation of single motion in chapter 4, we assumed that the structure in a medical image is primarily edges and only few corners. Thus, constraints on local motion, $\mathbf{c}$, is a representation that holds almost all relevant information in the image. In the case of multiple layers, we assume that the motion of each layer

can be described by motion constraints. For transparent layers, we also assume a sparse abundance of edges and that the small regions in the image are usually dominated by structure from only one layer. Under that assumption, it is possible to estimate constraints on the local motion. Each estimated motion constraint describes the motion of one layer, but we do not know which. An image can yield a million of motion constraints that need to be explicitly or implicitly clustered into multiple layers.

5.3 Correspondence Problems

We have already mentioned, there are correspondence problems when a large number of motion constraints are given from one image. Here follows a description in more detail about different types of correspondence problems.

5.3.1 Minimal Number of Motion Constraints

Generalized Aperture Problem[19]: Assume translation. In the case of one motion it is enough to see how two edges move, to estimate the motion. But, in the case of two transparent motions, we need at least five edges or independent motion constraints. Figure 5.1 illustrates that four motion constraints are never enough to estimate the motion of two layers due to ambiguous solutions. Adding a fifth constraint resolves the ambiguity, provided that one layer has three constraints and the other two layer has two constraints.

Theorem 5.3.1 *Assume there are M motion layers. The motion of each layer is represented by motion models with N parameters. Then we need at least $MN + M - 1$ motion constraints, of the type $c_x v_x + c_y v_y + c_t = 0$, to compute the motions of all the layers.*

Proof: There are N parameters per motion layer. Thus there are MN unknowns parameters. In addition, there are hidden unknowns telling which constraints belong to the same layer. Assume MN constraints are given, then there would be $\binom{MN}{N}$ points in parameter space where N constraints intersect. Only M of these correspond to a true motion. But if one extra extra constraint is added for all layers but one, i.e. a total of $M-1$ new constraints, then there are exactly $M-1$ points where $N+1$ constraints intersect. All of these $M-1$ points are true motions. The M:th motion is unambiguously given. As shown in figure 5.1, it is the only point of N constraints that remains after removal of the constraints belonging to the first $M-1$ motions.

So far, we have shown that $MN + M - 1$ motion constraints are enough. To complete the proof, we also must note that fewer constraints would cause ambiguities. If there are fewer constraints, there are two cases:
case I): One motion has $N-1$ constraint, or even fewer. It is impossible to solve for the N motion parameters. case II) At least two motions have fewer than $N+1$ constraints. The constraints of these motions will give $\binom{2N}{N}$ intersections of N

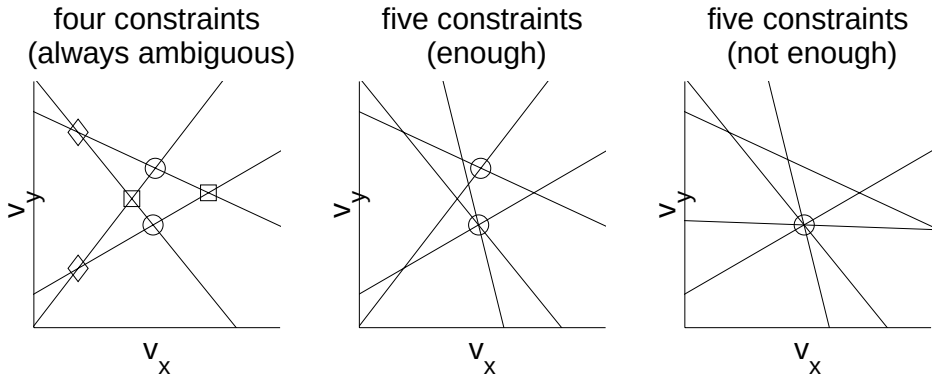

Figure 5.1: *To estimate two velocities, it is not enough to have four constraints, but five might be enough. The left figure shows four constraints and all possible solution cases. (Would you choose the circles, the squares or the diamonds?). With a fifth constraint, it is easy to realize that the circles represent the only solution. The third figure shows that five constraints are not always enough.*

constraints. If $N \geq 2$*, it is impossible to tell which of these correspond to the real motions.*

The problems is in fact worse than described here, since motion constraint vectors can be linearly dependent and noisy. Motion constraints will never intersect exactly at a number of points corresponding to each layer. In practice, the abundance of motion constraints will be denser at some points and it is hard to tell which constraint belongs to which layer.

5.3.2 Problem: Correspondence Between Estimates in Different Parts of the Image

Assume we have been able to locally estimate two or more motion vectors, $\tilde{\mathbf{v}}_1, \tilde{\mathbf{v}}_2, \ldots$, at every single pixel in an image. Then it remains to tell which motion vectors belong to the same layer or object. To illustrate the difficulties, we will study a case with ambiguous solutions. Figure 5.2 shows a field of two motion vectors

and two possible solutions of splitting up the vectors into two smooth fields. Of course, it is unlikely that two motion fields will be equal on a path across the image. Although, this will not happen in practice, we may get something quite close. In practice we will also have difficulties to tell if a motion field is continuous, due to noisy estimates and that motion is not given at points between pixels.

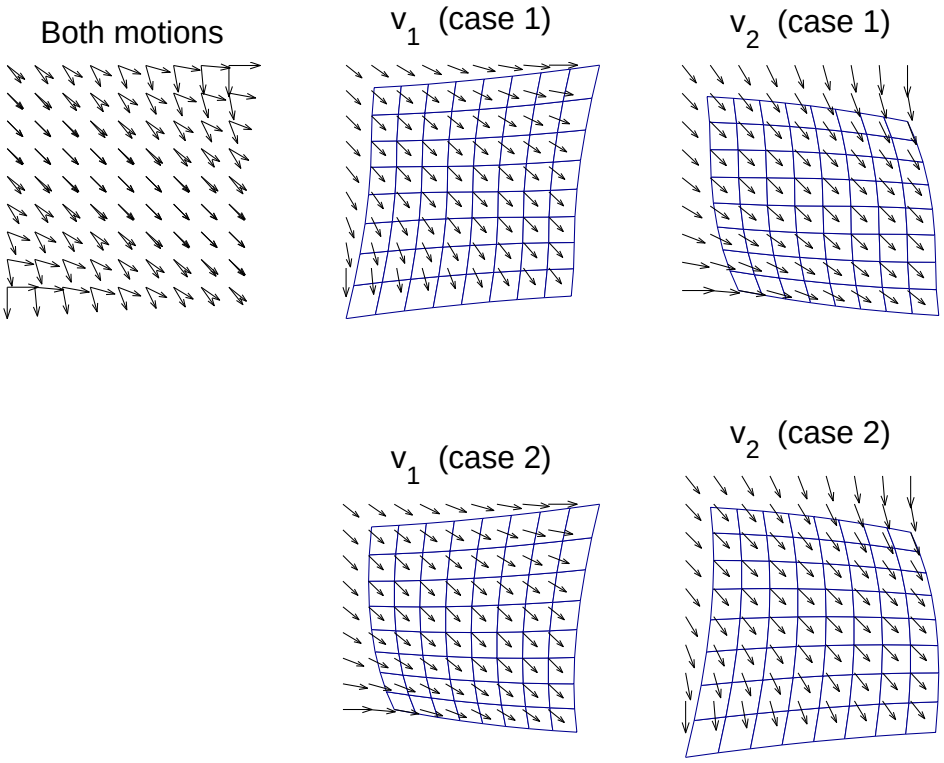

Figure 5.2: *Despite we have been able to estimate two velocity vectors at every point in the image, (upper left). We cannot unambiguously tell which velocity vectors belong to the same layer. In this case, there are two possible continuous solutions. Which one would you choose?*

5.3.3 Problem: Interframe Correspondence Between Estimates

Assume we have several frames. It is not enough to overcome all the previously mentioned problems, i.e. to estimate all motion vector fields for each frame. We still don't know which vector fields in the two frames correspond to the same layers. This might be a problem, even when motions are smooth over time. (We suggest that this problem should be solved by finding correspondence between the features in the images.)

Chapter 6

Estimation of Multiple Motions

Chapter 5 described the problems and difficulties in estimation of multiple motions. This chapter presents algorithms to overcome some of these problems and estimate motion fields of multiple layers. The primary focus is a modified version of the EM algorithm[19] for estimation of multiple motions.

6.1 Other Methods Considered

Before describing the successful part of our research, some other methods will be described briefly. We have considered a number of possible methods that we have decided not to use. Among them are explicit correlation and tracking of dominant layers. We cannot prove that they are inferior, but we describe problems that discourage further research.

6.1.1 Difficulties with Multiple Correlation Peaks

One of the methods we have considered is to explicitly correlate images with different shifts and find correlation peaks. As in estimation of a single motion, correlation is hard to extend to estimation of other motions than pure translations. Subpixel accuracy requires that the image is shifted with subpixel accuracy prior to correlation.

In estimation of multiple motions, it is often easy to find one of the motions as the highest peak. Finding the next peak is not as easy. It is like asking which is the second highest point in an area of mountains. Depending on who you ask, the answer is different One person may say it is a rock two meters below the highest peak. Another person would claim it is a minor peak of the same mountain, just hundred meters away. A third person would count nothing but another mountain, at least a kilometer away.

In a correlation map, the problem of defining criteria of finding a second peak

is as difficult as in the real world. It is eveb worse, since the correlation is only computed at a finite resolution of shifts. The limited resolution of the images make it unmeaningful to compute correlation for small subpixel shifts. If the difference between two layers is just a few pixels, it is likely that the two peaks merge into one. In order to estimate non-translational motions, local analysis is necessary and the second peak often drowns in the ridge of a higher peak.

6.1.2 Difficulties with Dominant Layers

Let's describe an approach that works in some of our experiments, but not good enough. It is based on the assumption that one layer may be much stronger than all the other layers. Under this assumption, we have been able to estimate motions of two transparent layers by first estimating motion of the dominant layer. Motion estimation is done using the phase-based method in section 4.2. The confidence measure is designed to suppress motions that are large relative to the warp. In conjunction with iterative refinement, section 4.2.3, the motion estimate converges to the dominant layer and outliers from the other layer are given low weight.

When motions of the dominant layer are known, it is possible to filter it away. The removal of the dominant layer from the images is far from perfect, but in our experiments it has been good enough for the next step. After removal of the dominant layer from the images, it is straightforward to estimate the motion of the weaker layer.

To improve accuracy, we have applied the above scheme iteratively. When both motions are known the two layers are separated, section 6.4. In the next iteration, the reconstructed layers are then used as reference images in the motion estimation. If success, the algorithm converges towards better reconstructed images and better motion estimates.

We have also been able to estimate motions in an image sequences where the layers are virtually equally strong. This was done using a bootstrap version of the above scheme. In the first integration, only two frames are used. Often, the motion estimate converges to either of the layers, although the accuracy is awfully bad. This layer is filtered out and used as a reference image in the motion estimation in the next iteration. Accuracy slowly gets better the more image frames that are used. The scheme is computationally expensive and suffers from problems with convergence. On our test images, it only works when motions are pure translations. It is also complicated to use multiple scales to estimate large motions since different layers are dominant at different scales.

6.2 Estimation of Motion Constraints

The motion constraints, $\mathbf{c}$ in this chapter are computed by the phase-based method we used for single motion estimation, section 4.2. It is possible to use other methods, but we have not tried that.

If a small region in the image only contains structure from one layer, the estimated motion constraint will be accurate. Otherwise, in case there is structure from two layers at the same point, they may interfere and produce outliers. The

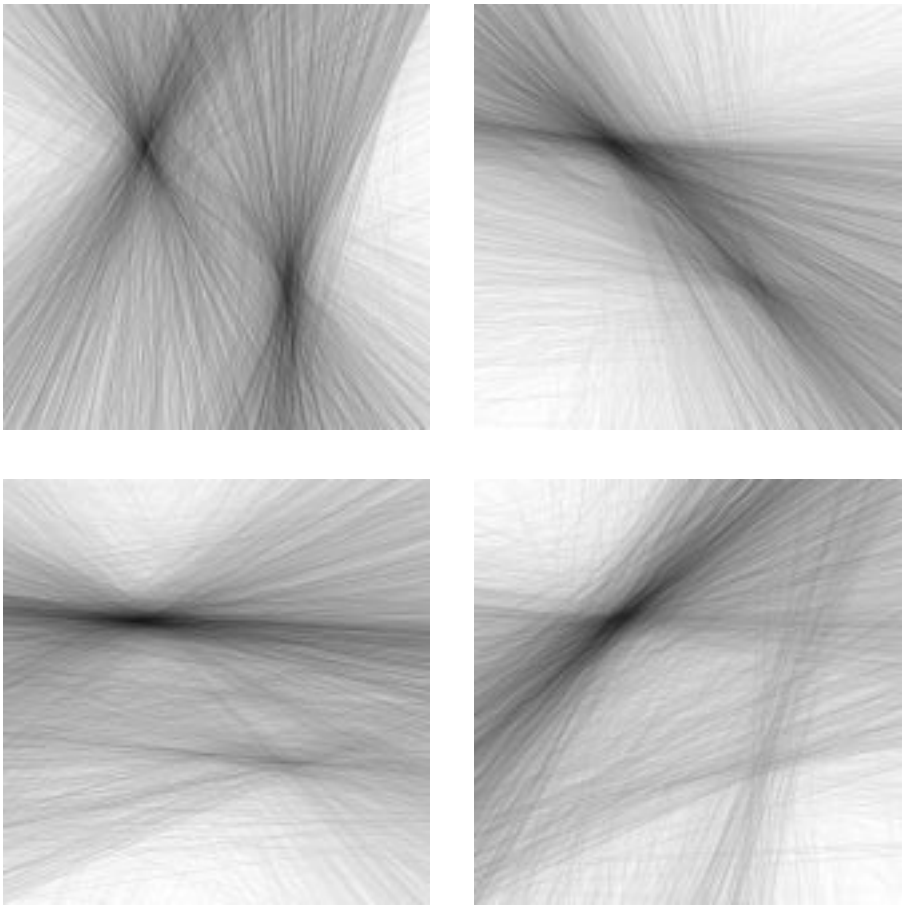

Figure 6.1: *Constraints from an image with two transparent layers as in section 6.5.3. Four directions of quadrature filters are used, yielding four constraints at each pixel. One layer appears stronger than the other.*

phase-based method is less susceptible to interference between layers than the conventional gradient method. The phase-based method is only sensitive for band pass frequencies and these are split up by in different directions, and thus dissimilar structure from different layers is less likely to interfere. The confidence measure is also designed to suppress matches of dissimilar structure. An example of constraints from two transparent layers is shown in figure 6.1.

6.3 EM (modified)

Out of the methods we have tried, the EM algorithm[19] is probably the best. The EM algorithm is a general algorithm with applications beyond imaging. In our

application, it is basically a kind of clustering algorithm, whose input is the mixture of all motion constraints from all layers. Motion constraints that are coherent are assumed to belong to the same layer. The EM algorithm is an iterative algorithm that uses an initial guess what the motions are and then does several iterations. A limitation is that the number of layers must be known a priori. Of course, no clustering algorithm for motion constraints is guaranteed to converge due to the correct answer, since there might be ambiguities as described in section 5.3.2. In addition, it happens that the EM algorithm gets stuck in a local optimum.

6.3.1 Review EM

When we have multiple motions, constraints intersect at different points in parameter space, corresponding to each of the motions. Estimating these motions is equivalent to finding the intersection points in parameter space. There seems to be no closed form solution to this problem, but it can be iteratively solved by the EM algorithm[19]. The EM algorithm is a clustering algorithm that iteratively applies two steps:

Expectation: Estimate the owner probabilities for each constraint, i.e. the probabilities that a constraint belongs to a particular motion layer. (We will see that the owner probabilities depend on previously made motion estimates.)

Maximization: Estimate the motions, when constraints are assigned to each of the motions, depending on the owner probabilities. Next iteration, the owner probabilities have changed since the motion estimates are different. As already mentioned, the original version of EM algorithm is only guaranteed to converge[26] to a local optimum but we do not know whether it is a global optimum.

6.3.2 Derivation of EM Algorithm for Multiple Warps

Jepson and Black [19] have used the the EM algorithm on multiple motions, but their approach didn't include warping images. As pointed out earlier, warping images is necessary to estimate large motions with best possible accuracy. This is especially important when estimating transparent motions. In case we wouldn't warp, the constraints of a large displacement would be much weaker than those of a small displacement.

The problem with warping multiple motions is that the image must be warped according to each of the estimated motions, producing multiple warped images. Here we will derive a simple extension of Jepson-Black's EM algorithm[19] that assigns different mixture probabilities to each of the warped images. A lot of variable names need to be introduced and it may help to keep an eye on the list in appendix B.

Let l denote the index of the warp. For each warp, l, we get a set of constraints $\mathbf{c}_{k,l}$, where k is a joint index of spatial position and other indices such as quadrature filter direction. Also assume the correct motions model parameters for each of

```
for i = 1 to number_of_iterations_refinement {
     for j = 1 to number_of_motions {
          warp_image;
          compute_motion_constraints;
     }
     for j = 1 to number_of_EM_iterations {
          E_step;
          M_step;
     }
}
```

Figure 6.2: *The loops in our extended EM algorithm with multiple warps.*

the motions are $\mathbf{a}_0, ..., \mathbf{a}_N$. Temporarily disregard from the possibility of a bad estimates of motion constraints. Under these conditions, the PDF[1] for observing the constraint $\mathbf{c}_{k,l}$ is

$$P(\mathbf{c}_{k,l}|\mathbf{x}_k, \{m_{n,l}\}, \mathbf{a}_0, ..., \mathbf{a}_N) = \sum_n m_{n,l} P(\mathbf{c}_{k,l}|\mathbf{x}_k, \mathbf{a}_n) \tag{6.1}$$

where $m_{n,l}$ is the probability of observing motion n in an image warped according to l. The PDF of observing our combination of constraints is the product of all PDFs for single constraints. By applying logarithm, the product is converted to a sum.

$$\begin{aligned} \log \prod_{k,l} P(\mathbf{c}_{k,l}|\mathbf{x}_k, \{m_{n,l}\}, \mathbf{a}_0, ..., \mathbf{a}_N) &= \sum_{k,l} \log P(\mathbf{c}_{k,l}|\mathbf{x}_k, \{m_{n,l}\}, \mathbf{a}_0, ..., \mathbf{a}_N) \\ &= \sum_{k,l} \log \sum_n m_{n,l} P(\mathbf{c}_{k,l}|\mathbf{x}_k, \mathbf{a}_n) \end{aligned} \tag{6.2}$$

We want to find the global maximum of this function under constraint that the mixture probabilities sum to 1.

$$\sum_n m_{n,l} = 1 \qquad \forall\, l = 1, 2, \ldots \tag{6.3}$$

6.3.3 Evaluating Criteria for Optimum

We have just arrived at a well defined mathematical problem, i.e. to maximize the joint PDF, eq. (6.2), under constraint eq. (6.3). To make clear what mathematical problem is, let's write the equations for an optimization problem. in a special form

$$\max_{\{\mathbf{a}_n\}, \{m_{n,l}\}} \sum_{k,l} \log \sum_n m_{n,l} P(\mathbf{c}_{k,l}|\mathbf{x}_k, \mathbf{a}_n) \tag{6.4}$$

$$\text{where} \quad \sum_n m_{n,l} - 1 = 0 \qquad \forall\, l \tag{6.5}$$

[1] Probability Density Function (PDF)

Similar to[19] we use Lagrange relaxation[2] to derive our version of the EM algorithm for warped images. Relaxation of eq. (6.5) gives the Lagrange function

$$L(\{\mathbf{a}_n\},\{m_{n,l}\},\{\mu_l\}) = \sum_{k,l} \log \sum_n m_{n,l} P(\mathbf{c}_{k,l}|\mathbf{x}_k,\mathbf{a}_n) - \sum_l \mu_l (\sum_n m_{nl} - 1) \tag{6.6}$$

where $\{\mu_l\}$ are the Lagrange multipliers. According to Lagrange theory, the optimum is a sadle point of $L(\{\mathbf{a}_n\},\{m_{n,l}\},\{\mu_l\})$ and must satisfy[3]

$$\frac{\partial}{\partial m_{n,l}} L(\{\mathbf{a}_n\},\{m_{n,l}\},\{\mu_l\}) = 0 \qquad \forall\, n,l \tag{6.7}$$

$$\frac{\partial}{\partial \mu_l} L(\{\mathbf{a}_n\},\{m_{n,l}\},\{\mu_l\}) = 0 \qquad \forall\, l \tag{6.8}$$

$$\nabla_{\mathbf{a}_n} L(\{\mathbf{a}_n\},\{m_{n,l}\},\{\mu_l\}) = \mathbf{0} \qquad \forall\, n \tag{6.9}$$

Evaluation of these equations yields

$$\sum_k \frac{P(\mathbf{c}_{kl}|\mathbf{x}_k,\mathbf{a}_n)}{\sum_{\bar{n}} m_{\bar{n}l} P(\mathbf{c}_{kl}|\mathbf{x}_k,\mathbf{a}_{\bar{n}})} - \mu_l = 0 \qquad \forall\, n,l \tag{6.10}$$

$$\sum_n m_{n,l} - 1 = 0 \qquad \forall\, l \tag{6.11}$$

$$\sum_{k,l} \frac{m_{nl} \nabla_{\mathbf{a}} P(\mathbf{c}_{k,l}|\mathbf{x}_k,\mathbf{a}_n)}{\sum_{\bar{n}} m_{\bar{n}l} P(\mathbf{c}_{kl}|\mathbf{x}_k,\mathbf{a}_{\bar{n}})} = \mathbf{0} \qquad \forall\, n \tag{6.12}$$

In order to further evaluate these equations, let's define something called ownership probabilities

$$q_{nkl} = \frac{m_{nl} P(\mathbf{c}_{k,l}|\mathbf{x}_k,\mathbf{a}_n)}{\sum_{\bar{n}} m_{\bar{n}l} P(\mathbf{c}_{kl}|\mathbf{x}_k,\mathbf{a}_{\bar{n}})} \tag{6.13}$$

Now, the equations can be written as

$$\sum_k q_{nkl} - \mu_l m_{nl} = 0 \qquad \forall\, n,l \tag{6.14}$$

$$\sum_n m_{nl} - 1 = 0 \qquad \forall\, l \tag{6.15}$$

$$\sum_{k,l} q_{nkl} \nabla_{\mathbf{a}} \log P(\mathbf{c}_{kl}|\mathbf{x}_k,\mathbf{a}_n) = \mathbf{0} \qquad \forall\, n \tag{6.16}$$

These are the equations that need to be satisfied at the optimum. They are solved iteratively by solving one at a time. Before describing the details in next section, we will give an intuitive meaning to the owner probabilities, q_{nkl}. Note that they are defined for each combination of motion constraint and layer. After a closer look at eq. (6.13), it is clear that q_{nkl} is the probability that the constraint $\mathbf{c}_{k,l}$ belongs to layer with index n. In particular, note that $\sum_n q_{nkl} = 1$.

[2] A common method in mathematics and optimization theory

[3] Unfortunately, even a local optimum satisfies these equations.

6.3.4 Iterative Search for Optimum

The EM algorithm defines how to solve equations. (6.14)-(6.16) iteratively by solving one variable at a time using one equation.

The first operation in each iteration is to compute the ownership probabilities for each pixel and layer. This is a straightforward computation using eq. (6.13). In the first iteration, we need an initial guess of the motion parameters, $\{\mathbf{a}_n\}$, and mixture probabilities, $\{m_{nl}\}$.

The second operation is to compute the motion parameters for each layer, $\{\mathbf{a}_n\}$ using eq. (6.16). Thanks to our probability function that will be defined in section 6.3.5, the motion estimation is the same least square fit as in section 3.2.

In order to prepare for next iteration, the mixture probabilities, $\{m_{nl}\}$, need to be updated using

$$m_{nl} = \frac{\sum_k q_{nkl}}{\sum_{\bar{n},k,\bar{l}} q_{\bar{n}k\bar{l}}}. \tag{6.17}$$

Then we go back and do some more iterations. The EM algorithm is guaranteed to converge to a local optimum[26].

6.3.5 The Probability Function

The probability density function, defines the probability of observing a particular constraint at a particular spatial location, according to a particular motion model.

For simplicity, we use a Gaussian PDF. It is simple because eq. (6.16) yields the same equations as for the model based estimation in the section 3.2, except for that no confidence measure is used. In next section we will tell how to get the confidence measure back.

The probability of observing a particular constraint is a normal distribution with respect to the deviation according to a dissimilarity measure[19], $d(\mathbf{c}, \mathbf{v})$.

$$P(\mathbf{c}|\mathbf{x}, \mathbf{a}) = P(\mathbf{c}|\mathbf{v}) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{d^2(\mathbf{c}, \mathbf{v})}{2\sigma^2}) \tag{6.18}$$

where $\mathbf{v} = \mathbf{K}(\mathbf{x})\,\mathbf{a}$. With some abuse of notation, we define the d-function

$$d^2(\mathbf{c}, \mathbf{v}) = \frac{(c_x v_x + c_y v_y + c_t)^2}{c_x^2 + c_y^2} \tag{6.19}$$

The denominator acts like normalizing the vector (c_x, c_y) and the value of $d(\mathbf{c}, \mathbf{v})$ is the closest distance from the point (v_x, v_y) to the line $c_x v_x + c_y v_y + c_t = 0$. Note that our function does not assume larger deviations for large motions in contrast to[19]. Our approach to warp the image gives the same absolute accuracy for arbitrary large motions.

6.3.6 Introducing Confidence Measure in the EM Algorithm

In the probability function defined in section 6.3.5, the confidence measure was removed due to the normalization in eq. (6.19). We believe the confidence should

not affect the relative values of the owner probabilities. For example, a high confidence in the motion constraint does not mean that we are certain which layer it belongs to.

Without writing the equations again, we will tell how to derive the EM algorithm with a confidence measure. Let $C = c_x^2 + c_y^2$ denote the confidence measure of a motion constraint. The confidence measure should be introduced in eq. (6.4) by multiplying $C_{k,l}^2$ in the outer sum in front of "log...". This confidence will follow through all the derivation in section 6.3.3. In the end, eq. (6.14) and eq. (6.16) will be modified by simply replacing q_{nkl} with $C_{k,l}^2\, q_{nkl}$.

6.3.7 Our Extensions to the EM Algorithm

As we have pointed out, it is proved that the EM algorithm converges to a local optimum, but the risk of getting stuck in a local optimum is prohibitive when using motion models with many degrees of freedom. In our case, warping images, makes convergence even more hazardous, since the image has finite size and we might get outside the boundary. In experiments, when applying the EM algorithm to an images with two transparent layers with affine motion model, the EM algorithm sometimes bails out already in the first iteration. Our remedy to this is to control the stiffness (section 3.3). In the first iteration, only translations are estimated. In the second iteration, we allow small affine deformations. For every iteration, we reduce the cost. We do several EM iterations for every time we warp the image.

Other researchers[21] suggest simulated and deterministic annealing to avoid getting stuck in a local optimum. This would require too many iterations when we have many parameters. We have not tried that since we have not had problems with local optima when using cost functions.

Another extension is to let owner probabilities alter the certainties. If we are not sure which layer a constraint belongs to, its influence in estimation of the motion parameters should be less.

Outliers in motion constraints are more frequent when there are multiple layers, since structures corresponding to different layers sometimes interfere. In our scheme, outliers are handled by introducing an extra layer of motion that is supposed to own outliers. This special layer has a special probability function that is much wider than for the other layers. This means that this layer owns all constraints that are far away from the closest estimated motion. How far is implicitly determined by specifying a value for the mixture probability, i.e. we want that

$$m_{n,N} = \gamma \quad \forall n \tag{6.20}$$

where N is the motion model and γ is a predefined constant, that controls what fraction of constraints to consider as outliers.

This is controlled by setting

$$P(\mathbf{c}_{k,l}|\mathbf{x}_k, \mathbf{a}_N) = p_{outlier} \tag{6.21}$$

$p_{outlier}$ is determined so that eq. (6.20) holds.

Since this probability function does not depend on its corresponding motion, there is no need to estimate the motion parameters of this layer.

6.3.8 Convergence of Modified EM with Warp

We have made we have made modifications of the EM algorithm without proving convergence. Even if we assumes that the modified EM algorithm always converges, it would not imply that the iterative refinement with image warps converges.

It is important not to confuse the iterations of the EM algorithm with the iterations of image warps. The EM algorithm is in the inner loop and the warps are in the outer loop (see figure 6.2. For every iteration of iterative refinement, several EM iterations are performed. The proof of convergence has nothing to do with convergence of iterative refinement. Convergence of the inner loop does not imply convergence of the outer loop.

6.4 Reconstruction of Transparent Layers

If motions are known it is possible to reconstruct transparent layer, except for the very lowest frequencies and provided that motions are unique and big enough not to interfere with the pixel resolution. A predecessor of our algorithm is to simply average along the trajectory of one motion[18]. If we have many frames in the image sequence, the structure corresponding to this motion is sharpened and all other structure is blurred. We improve the image quality by estimating the errors and feeding them back. We arrive at an iterative backprojection algorithm, described by figure 6.3.

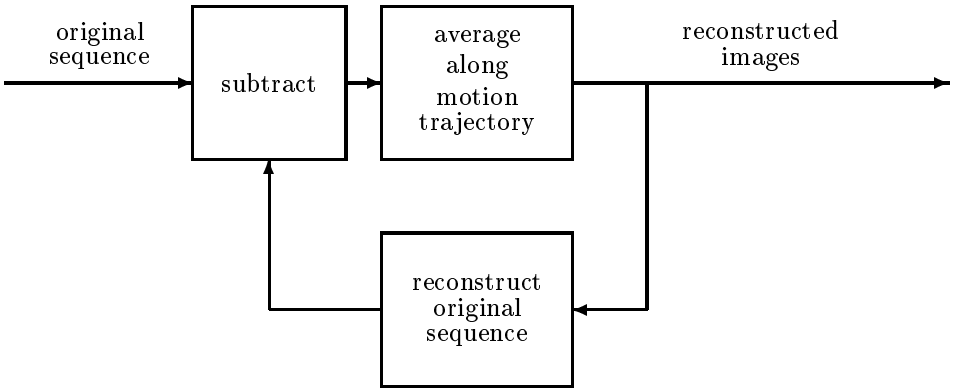

Figure 6.3: *Reconstruction of motion layers using simple backprojection.*

6.4.1 Improved Backprojection Algorithm

In the simple backprojection, the feedback images are warped two times; first when feeded back and then when feeded forward after being subtracted. The double warp degrades the image quality. Our way to overcome this problem is the

scheme in figure 6.4 where the two warps are performed in one step as a single warp.

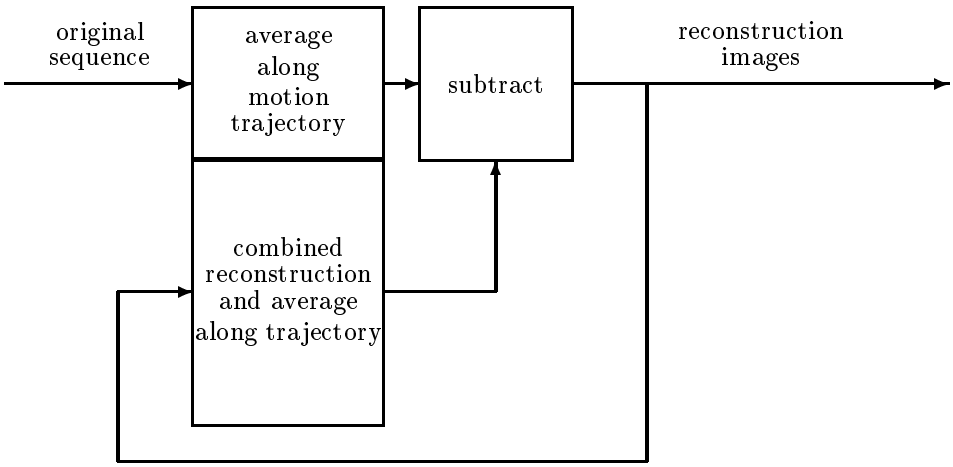

Figure 6.4: *Reconstruction of transparent layers using a more sophisticated back-projection.*

6.4.2 Finding Correspondence between Motion Estimates from Different Frames

As pointed out in section 5.3, we need know which motion vectors correspond over time. The way we have applied the EM algorithm does not give us that information. Comparing motion vectors over time does not work since our motions are irregular over time. Our approach is to first reconstruct layers from only two frames. (There are no such problems if we only have two frames). Although image quality is bad, we have the two layers separated and we can analyze how these correlates with frames later in the sequence (when warped with different motion estimates).

6.4.3 Experimental Results

So far, we have run the our algorithms only on images with two layers that have been superimposed synthetically. Figure 6.5 shows a number of frames from a sequence of synthetically generated images. The images have been generated by taking two still images on a heart and adding them together with affine random motions and then crop the valid region. The original heart images were thrown away and the only input data to our algorithm was the sequence of 50 generated

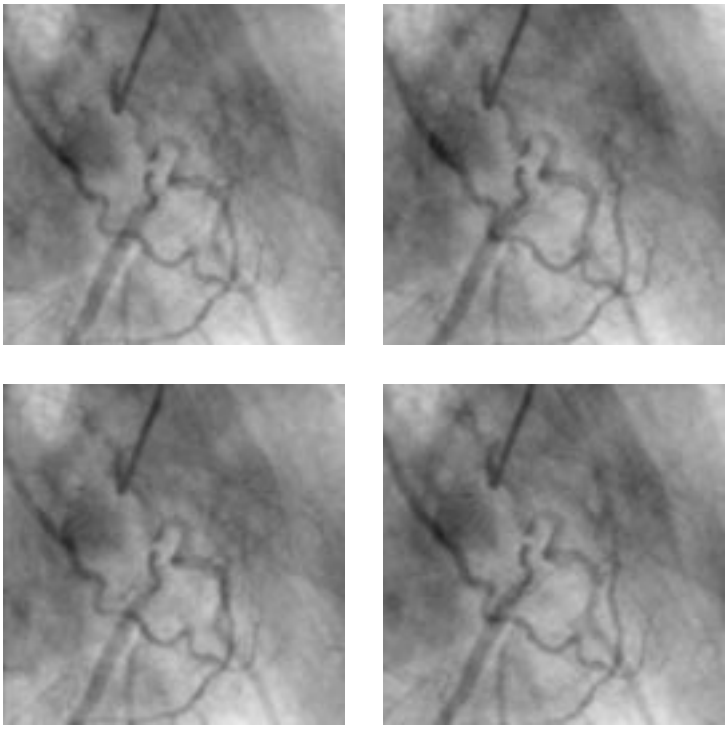

Figure 6.5: *Synthetic multiple motion field sequence containing two layers with independent random affine motions. Frame 10, 20, 30 and 40 in a sequence of 50 images.*

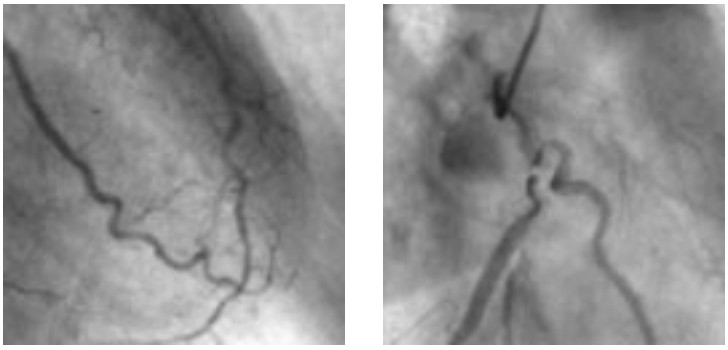

Figure 6.6: *Reconstructed images (cropped a few pixels to hide artifacts at borders). Compare to figure 1.3.*

images. Using the EM algorithm with our phase-based method, the motions of both layers were estimated under assumption of affine motions. The original layers were reconstructed and shown in figure 6.6. We get some artifacts at the borde is degraded.

We did not save the true original images are thrown away in order to avoid confusion, but they can be seen in figure 1.3. One layer is the upper left part of one image in figure 1.3 and the other layer is the lower right part. The reconstruction of the details is fine but the lowpass is degraded and the DC is discarded.

After doing these experiments, we have seen a poster and an abstract on a project that also seem to solve the same problem of separation of layers. Not much details related to our work are given and the proceedings with the full article[28] is not yet available.

6.5 Alternative Method for Two Mixed Motions

In this section, we present an alternative to the EM algorithm for estimation of multiple motions. Compared to the EM algorithm, the same input data is used, but the computational time is usually shorter since it is cheap to do many iterations once some initial computations are done. Among the drawbacks of this method is the influence of outliers seem too large, problems with convergence and we have not yet invented any method to take advantage of multiple warps in order to estimate large motions with good accuracy.

We have found references of an algorithm[30, 31] with some similarities. It uses higher order moments in 3D Fourier/Gabor transforms of spatiotemporal volumes and also yields a minimization problem.

6.5.1 Basic Idea

Assume we have two motions, $\mathbf{v}_1$ and $\mathbf{v}_2$, described by parameter vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ for some motion model defined by $\mathbf{K}(\mathbf{x})$. As defined in chapter 3,

$$\mathbf{v}_1 = \mathbf{K}(\mathbf{x})\,\mathbf{a}_1 \quad \text{and} \quad \mathbf{v}_2 = \mathbf{K}(\mathbf{x})\,\mathbf{a}_2 \tag{6.22}$$

A large number of constraints vectors, $\mathbf{c}_k$, $k = 1, 2, 3, \ldots$ are given at spatial positions $\mathbf{x}_k$. Neglecting interference between layers, the motion constraints are supposed to satisfy either

$$\mathbf{c}_k^T \overline{\mathbf{v}}_1 = 0 \quad \text{or} \quad \mathbf{c}_k^T \overline{\mathbf{v}}_2 = 0 \tag{6.23}$$

where

$$\overline{\mathbf{v}} = \begin{pmatrix} \mathbf{v} \\ 1 \end{pmatrix} \quad \text{and} \quad \mathbf{c}_k = \begin{pmatrix} c_{k,x} \\ c_{k,y} \\ c_{k,t} \end{pmatrix}. \tag{6.24}$$

Let's now define an error measure similar to eq. (3.12) but for two motions,

$$\begin{aligned}\varepsilon(\mathbf{a}_1, \mathbf{a}_2) &= \sum_k \left(\mathbf{c}_k^T \, \overline{\mathbf{v}}_1(\mathbf{x}_k)\right)^2 \left(\mathbf{c}_k^T \, \overline{\mathbf{v}}_2(\mathbf{x}_k)\right)^2 \\ &= \sum_k \left(\begin{pmatrix} c_{k,x} & c_{k,y} \end{pmatrix} \mathbf{K}(\mathbf{x}_k) \, \mathbf{a}_1 + c_{k,t}\right)^2 \left(\begin{pmatrix} c_{k,x} & c_{k,y} \end{pmatrix} \mathbf{K}(\mathbf{x}_k) \, \mathbf{a}_2 + c_{k,t}\right)^2\end{aligned} \tag{6.25}$$

To simplify notations, we introduce

$$\mathbf{b_k} = \begin{pmatrix} c_{k,x} & c_{k,y} \end{pmatrix} \mathbf{K}(\mathbf{x}_k) \tag{6.26}$$

$$d_k = c_{k,t} \tag{6.27}$$

and the expression gets more readable,

$$\begin{aligned}\varepsilon(\mathbf{a}_1, \mathbf{a}_2) &= \sum_k (\mathbf{b}_k \mathbf{a}_1 + d_k)^2 \left(\mathbf{b}_k \mathbf{a}_2 + d_k\right)^2 \\ &= \sum_k (\mathbf{a}_1^T \mathbf{b}_k \mathbf{b}_k^T \mathbf{a}_1 + 2 d \mathbf{b}_k^T \mathbf{a}_1 + d_k^2)(\mathbf{a}_2^T \mathbf{b}_k \mathbf{b}_k^T \mathbf{a}_2 + 2 d_k \mathbf{b}_k^T \mathbf{a}_2 + d_k^2).\end{aligned} \tag{6.28}$$

After further evaluation of the product above, it turns out that the sum can be moved inside the unknown parameter vectors, $\mathbf{a}_1$ and $\mathbf{a}_2$. We have to sum over outer products of up to four vectors and we get a four dimensional array of numbers, that are called tensors[34]. Readers not familiar of tensors, can think of them as an extension of vectors and matrices into arbitrary dimensionality. Tensors come with special tensor notations, where matrix-like product are written without Σ. Instead indices to multiply and sum over are written as subscripts of one factor and subscript of the other factor.

$$\begin{aligned}\varepsilon(\mathbf{a}_1, \mathbf{a}_2) = & T_4^{ijkl} a_{1i} a_{1j} a_{2k} a_{2l} + \\ & + T_3^{ijk} a_{1i} a_{1j} a_{2k} + T_3^{ijk} a_{1i} a_{2j} a_{2k} + \\ & + T_2^{ij} a_{1i} a_{1j} + 4 T_2^{ij} a_{1i} a_{2j} + T_2^{ij} a_{2i} a_{2j} + \\ & + 2 T_1^i a_{1i} + 2 T_1^i a_{2i} + \\ & + T_0\end{aligned} \tag{6.29}$$

where the T_4^{ijkl}, T_3^{ijk}, T_2^{ij}, T_1^i, T_0 are tensors with 4, 3, 2, 1, 0 indices[4]. The tensors are formed by summing outer products[5] and we get the fourth moments

[4] Example: T_2 has two indices and is a matrix, T_1 has one index and is a vector. T_0 has no index and hence a scalar.

[5] For example, the outer product of two column vectors, $\mathbf{u} \otimes \mathbf{v} = \mathbf{u} \, \mathbf{v}^T$ is a matrix (or a tensor with two indices).

of the elements in $\mathbf{c}$

$$\mathbf{T}_4 = \sum_k \mathbf{b}_k \otimes \mathbf{b}_k \otimes \mathbf{b}_k \otimes \mathbf{b}_k \tag{6.30}$$

$$\mathbf{T_3} = \sum_k d_k \mathbf{b}_k \otimes \mathbf{b}_k \otimes \mathbf{b}_k \tag{6.31}$$

$$\mathbf{T_2} = \sum_k d_k^2 \mathbf{b}_k \otimes \mathbf{b}_k \tag{6.32}$$

$$\mathbf{T_1} = \sum_k d_k^3 \mathbf{b}_k \tag{6.33}$$

$$\mathbf{T_0} = \sum_k d_k^4 \tag{6.34}$$

6.5.2 Minimizing $\varepsilon(\mathbf{a}_1, \mathbf{a}_2)$

In order to to find motion, $\varepsilon(\mathbf{a}_1, \mathbf{a}_2)$ is minimized. In lack of references for better methods, a version of Newton's method is used to find a stationary points, i.e. where the gradient is zero. A brief description of the approach would be a multidimensional version of the well known Newton-Raphson method applied on the gradient. The gradient with respect to $\mathbf{a}_1$ is computed as

$$\begin{aligned}\nabla_{\mathbf{a}_1}\varepsilon(\mathbf{a}_1, \mathbf{a}_2) = T_4^{ijkl} a_{1i} a_{2j} a_{2k} + \\ + 4T_3^{ijk} a_{1i} a_{2j} + 4T_3^{ijk} a_{2i} a_{2j} + \\ + 2T_2^{ij} a_{1i} + 4T_2^{ij} a_{2i} + \\ + 2T_1\end{aligned} \tag{6.35}$$

and the gradient with respect to $\mathbf{a}_2$ is computed in a similar way. With some abuse of tensor notations, we treat the tensors as vectors and say that the gradient $\nabla\varepsilon = \begin{pmatrix}\nabla_{\mathbf{a}_1}\varepsilon(\mathbf{a}_1, \mathbf{a}_2) \\ \nabla_{\mathbf{a}_2}\varepsilon(\mathbf{a}_1, \mathbf{a}_2)\end{pmatrix}$. With similar abuse of notations, the Hessian, i.e. matrix of second derivatives, is given by

$$\begin{aligned}\nabla^2\varepsilon(\mathbf{a}_1, \mathbf{a}_2) = \begin{pmatrix} 2T_4^{ijkl} a_{2i} a_{2j} & 4T_4 a_{1i} a_{2j} \\ 4T_4 a_{1i} a_{2j} & 2T_4^{ijkl} a_{1i} a_{1j}\end{pmatrix} + \\ + \begin{pmatrix} 4T_3^{ijk} a_{2i} & 4T_3 ijk a_{1i} + 4T_3 ijk a_{2i} \\ 4T_3 ijk a_{1i} + 4T_3 ijk a_{2i} & 4T_3^{ijk} a_{1i}\end{pmatrix} + \\ + \begin{pmatrix} 2T_2^{ij} & 4T_2^{ij} \\ 4T_2^{ij} & 2T_2^{ij}\end{pmatrix}\end{aligned} \tag{6.36}$$

The parameter vectors, $\mathbf{a}_1, \mathbf{a}_2$ are computed iteratively using Newton's method. (Search for stationary points)

$$\begin{pmatrix}\mathbf{a}_1 \\ \mathbf{a}_2\end{pmatrix}^{(n+1)} = \begin{pmatrix}\mathbf{a}_1 \\ \mathbf{a}_2\end{pmatrix}^{(n)} - \left(\nabla^2\varepsilon(\begin{pmatrix}\mathbf{a}_1 \\ \mathbf{a}_2\end{pmatrix}^{(n)})\right)^{-1} \nabla\varepsilon(\begin{pmatrix}\mathbf{a}_1 \\ \mathbf{a}_2\end{pmatrix}^{(n)}) \tag{6.37}$$

Convergence is often quite poor. In our experiments, the Newton search often converges to suboptimal solutions, usually $\mathbf{a}_1 = \mathbf{a}_2$. Our simple remedy to this problem is to use several start points for the iteration. The Newton search is so fast[6] so that we can use hundreds of start points. A Newton search that tends to bail out is canceled and we try the next start points. The procedure is repeated until we have got a large number of sound estimates of $\mathbf{a}_1$ and $\mathbf{a}_2$. Then we choose the estimate with the smallest $\varepsilon(\mathbf{a}_1, \mathbf{a}_2)$. Unfortunately, we can never be sure that we have found the optimum. All we can do is to increase the likelihood by using a large number of start points.

An alternative to this approach is simulated or deterministic annealing. Annealing also suffers from the problem that you cannot be sure you have found the optimum.

6.5.3 Experimental Results

This alternative method has been implemented both for translational and affine motions. Accuracy seems not as good as the EM algorithm. Figure 6.7 shows results from experiments on synthetic images. Just like in figure 4.6 in chapter 4, the phase-based method is used to estimate constraints and the image is not warped to improve accuracy. The same test images are used(Lena+Debbie128) but these are superimposed with opposite motion. When estimating multiple motions, we get two motion estimates for each pixel and it is hard to tell which corresponds to which layer. In evaluation of accuracy, the motion estimates are sorted by a comparison with the known motion and we don't think it's cheating. Motion constraints for this experiment when motions are (1, 1) pixels in each direction are drawn in figure 6.1.

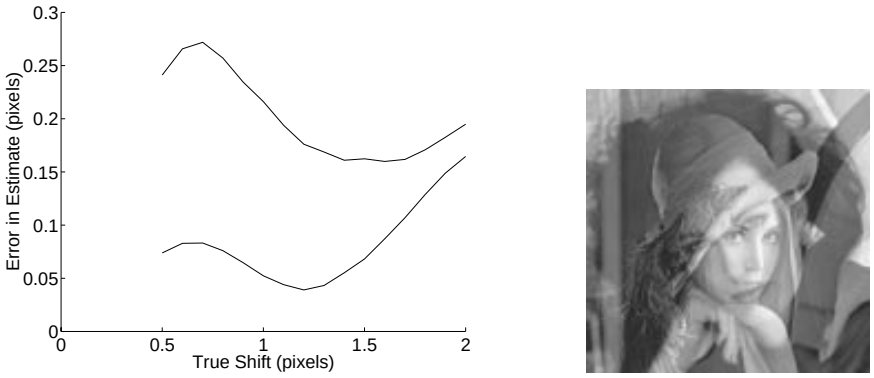

Figure 6.7: *Accuracy of estimation of each of two superimposed layers. For small motions, it is hard to separate the layers. Evidently, one layer yields better estimates than the other.*

[6] Very fast compared to the EM algorithm

Chapter 7

Canonical Correlation of Complex Variables.

There is a well developed theory for canonical correlation analysis (CCA) of real variables, Borga[5]. Canonical correlation of complex variables has successfully been used in a stereo algorithm[5] without having a theory for the complex case. This chapter introduces a novel way of maximizing canonical correlation, which is derived for complex variables. It is also shown to generate the same solution as Borga's[5] method, even for complex variables. Thus, Borga's method is proven to work even for the complex case. A major advantage of our novel method is the ability to handle singular covariance matrices.

This chapter is a theoretical study on canonical correlation in general. Since no images or vectors are involved, a number of variable names and notations can be used for other purposes. For example, vector $\mathbf{v}$ is not the motion vector.

For complex matrices, conjugate and transpose are usually applied simultaneously. This is denoted by superscript star($*$), e.g. $\mathbf{A}^*$. A simple transpose is denoted by superscript T, e.g. $\mathbf{A}^T$. Unfortunately, this chapter uses simple complex conjugate without transpose. In lack of good notations, a simple conjugate is written as a combination of a star and transpose, e.g. $\mathbf{A}^{T*}$.

Another commonly used notation is the operator of expectancy value of a stochastic variable, $E[.]$. In practical application, statistical data sets are limited and we need to use estimates of expectancy value. After having verified all the formulas in the chapter, it turns out that every every $E[.]$ operator can be substituted with a sum over all available data.

7.1 Definition of Canonical Correlation of Complex Variables

The notations and formulas are similar to Borga's PhD thesis[5], except for some variables names that would cause too much confusion in image processing. Assume we have two sets of stochastic variables organized in two vectors, $\mathbf{z}_A$ and $\mathbf{z}_B$

respectively. For each of the two vectors we construct linear combinations of the vector components.

$$z_A = \mathbf{w}_A^T \mathbf{z}_A \quad \text{and} \quad z_B = \mathbf{w}_B^T \mathbf{z}_B \tag{7.1}$$

where $\mathbf{w}_A$ and $\mathbf{w}_B$ are vectors of linear combination coefficients. The canonical correlation is the correlation of these two linear combinations.

$$\begin{aligned}
\rho &= \frac{E[z_A^* z_B]}{\sqrt{E[z_A^* z_A]E[z_B^* z_B]}} \\
&= \frac{E[(\mathbf{z}_A^T \mathbf{w}_A)^* \mathbf{z}_B^T \mathbf{w}_B]}{\sqrt{E[(\mathbf{z}_A^T \mathbf{w}_A)^* \mathbf{z}_A^T \mathbf{w}_A]E[(\mathbf{z}_B^T \mathbf{w}_B)^* \mathbf{z}_B^T \mathbf{w}_B]}} \\
&= \frac{\mathbf{w}_A^* E[\mathbf{z}_A^{T*} \mathbf{z}_B^T] \mathbf{w}_B}{\sqrt{\mathbf{w}_A^* E[\mathbf{z}_A^{T*} \mathbf{z}_A^T] \mathbf{w}_A \, \mathbf{w}_B^* E[\mathbf{z}_B^{T*} \mathbf{z}_B^T] \mathbf{w}_B}} \\
&= \frac{\mathbf{w}_A^* \mathbf{C}_{AB} \mathbf{w}_B}{\sqrt{\mathbf{w}_A^* \mathbf{C}_{AA} \mathbf{w}_A \, \mathbf{w}_B^* \mathbf{C}_{BB} \mathbf{w}_B}}
\end{aligned} \tag{7.2}$$

where the covariance[1] matrices are

$$\mathbf{C}_{AA} = E[\mathbf{z}_A^{T*} \mathbf{z}_A^T] \quad \text{and} \quad \mathbf{C}_{AB} = E[\mathbf{z}_A^{T*} \mathbf{z}_B^T] \quad \text{and} \quad \mathbf{C}_{BB} = E[\mathbf{z}_B^{T*} \mathbf{z}_B^T] \tag{7.3}$$

and $\mathbf{w}_A$ and $\mathbf{w}_B$ are computed to maximize the correlation.

7.2 Maximizing Canonical Correlation

The objective in canonical correlation analysis is to find the two linear combinations that yield maximum correlation, i.e. maximizing the correlation, ρ, with respect to $\mathbf{w}_A$ and $\mathbf{w}_B$. In the complex case, where ρ is complex, the first issue is what to maximize, the absolute value or the real part. The following theorem implies that both the absolute value and the real part can be maximized simultaneously.

Theorem 7.2.1

$$\max \Re \rho = \max |\rho| \tag{7.4}$$

Proof: It is obvious that $\max \Re\rho \leq \max \|\rho\|$. *It remains to show that* $\max \Re\rho \geq \max \|\rho\|$ *always holds. Assume that we find* $\mathbf{w}_A$ *and* $\mathbf{w}_B$ *so that* $\|\rho\|$ *is maximized but* $\arg \rho \neq 0$. *Then we can get a real canonical correlation with the same absolute value by multiplying* $\mathbf{w}_A$ *by* $e^{i \arg \rho}$.

At maximum, the linear combination coefficients hold information about dependence of the input data. In learning and adaptive filtering, these linear combination can be applied on new input data for classification. The stereo and motion algorithm in chapter 8 use analysis of $\mathbf{w}_A$ and $\mathbf{w}_B$ directly to find mutual dependence between the two images. A simple example of canonical correlation analysis is provided in appendix A.2.

[1] Only true covariance if expectancy value is zero.

7.3 Properties of the Canonical Correlation

Theorem 7.3.1

$$\rho = \frac{\mathbf{w}_A^* \mathbf{C}_{AB} \mathbf{w}_B}{\sqrt{\mathbf{w}_A^* \mathbf{C}_{AA} \mathbf{w}_A \, \mathbf{w}_B^* \mathbf{C}_{BB} \mathbf{w}_B}} \leq 1 \tag{7.5}$$

Proof: Note that $E[z_A^* z_B]$ is a scalar product[2] of stochastic variables z_A and z_B. Thus, it follows from Cauchy-Schwarz' that the numinator is less or equal to the denuminator. In real world applications, expectancy value, $E[z_A^* z_B]$, is substituted with a sum of all available data, $\sum_k z_{A,k}^* z_{B,k}$. This sum also meets the criteria for being a scalar product. Thus, it still holds that $|\rho| \leq 1$.

7.4 Maximization Using SVD

Borga[5] transforms the maximization problem of canonical correlation into a generalized eigenvector problem. The formulas on page 68 in his dissertation[5] are only formulated for real parameter vectors, $\mathbf{w}_A$ and $\mathbf{w}_B$. That proof does not hold for the complex case, since it is not possible to compute the derivative of a complex conjugate (see note in appendix A.1). It may be possible to modify Borga's proof by differentiating with respect to real and imaginary parts separately, but we do not present any such proof. Instead, we present a novel proof and a novel method that employs neither derivatives nor generalized eigenvector problem.

Our novel method of maximizing the canonical correlation, works, unlike the scheme by Borga[5], even when covariance matrices $\mathbf{C}_{AA}$ and $\mathbf{C}_{BB}$ are singular. We will also show that it is equivalent to Borga's method, even for the complex case of canonical correlation. Thus, we have proved that Borga's method is valid for complex variables.

7.4.1 Operations in Maximization

Since $\mathbf{C}_{AA}$ and $\mathbf{C}_{BB}$ are Hermitian and positive definite, we can do eigenvalue decomposition

$$\mathbf{C}_{AA} = \mathbf{Q}_A \mathbf{D}_A^2 \mathbf{Q}_A^* \quad \text{and} \quad \mathbf{C}_{BB} = \mathbf{Q}_B \mathbf{D}_B^2 \mathbf{Q}_B^* \tag{7.6}$$

where $\mathbf{Q}_A$ and $\mathbf{Q}_B$ are unitary[3] matrices. $\mathbf{D}_A$, $\mathbf{D}_B$ are diagonal matrices, whose eigenvalues are real and nonnegative. Note that one or more eigenvalues are zero in case $\mathbf{C}_{AA}$ or $\mathbf{C}_{BB}$ is singular. In practice, matrices are almost never exactly singular, just ill conditioned. Therefore, it may be necessary to threshold eigenvalues in $\mathbf{D}_A$ and $\mathbf{D}_B$.

Define

$$\mathbf{v}_A = \mathbf{D}_A \mathbf{Q}_A^* \mathbf{w}_A \quad \text{and} \quad \mathbf{v}_B = \mathbf{D}_B \mathbf{Q}_B^* \mathbf{w}_B \tag{7.7}$$

[2] A scalar product in complex vector space must conjugate one of the factors.
[3] A matrix, $\mathbf{Q}$ is unitary if its inverse is $\mathbf{Q}^*$, i.e. $\mathbf{Q}^*\mathbf{Q} = \mathbf{I}$.

which is a conventional coordinate transformation in the nonsingular case. In the singular case, one or more elements in $\mathbf{v}_A$ or $\mathbf{v}_B$ are always zero. Let's also define a covariance matrix for this coordinate transformation.

$$\tilde{\mathbf{C}}_{AB} = \mathbf{D}_A^\dagger \mathbf{Q}_A^* \mathbf{C}_{AB} \mathbf{Q}_B \mathbf{D}_B^\dagger \tag{7.8}$$

where $\dagger$ denotes pseudo inverse[4].

With this coordinate transformation, the canonical correlation can be expressed in a simple form. Thanks to the relations between $\mathbf{C}_{AA}$, $\mathbf{C}_{AB}$ and $\mathbf{C}_{BB}$, the following equations are valid even when $\mathbf{D}_A$ and $\mathbf{D}_B$ are singular. For readability, this proof is put in appendix A.3.

$$\begin{aligned}
\rho &= \frac{\mathbf{w}_A^* \mathbf{C}_{AB} \mathbf{w}_B}{\sqrt{\mathbf{w}_A^* \mathbf{C}_{AA} \mathbf{w}_A \; \mathbf{w}_B^* \mathbf{C}_{BB} \mathbf{w}_B}} \\
&= \frac{\mathbf{w}_A^* \mathbf{Q}_A \mathbf{Q}_A^* \mathbf{C}_{AB} \mathbf{Q}_B \mathbf{Q}_B^* \mathbf{w}_B}{\sqrt{\mathbf{w}_A^* \mathbf{Q}_A \mathbf{D}_A^2 \mathbf{Q}_A^* \mathbf{w}_A \; \mathbf{w}_B^* \mathbf{Q}_B \mathbf{D}_B^2 \mathbf{Q}_B^* \mathbf{w}_B}} \\
&\quad \text{see appendix A.3} \\
&= \frac{\mathbf{w}_A^* \mathbf{Q}_A \mathbf{D}_A \, \mathbf{D}_A^\dagger \mathbf{Q}_A^* \mathbf{C}_{AB} \mathbf{Q}_B \mathbf{D}_B^\dagger \, \mathbf{D}_B \mathbf{Q}_B^* \mathbf{w}_B}{\sqrt{\mathbf{w}_A^* \mathbf{Q}_A \mathbf{D}_A^2 \mathbf{Q}_A^* \mathbf{w}_A \; \mathbf{w}_B^* \mathbf{Q}_B \mathbf{D}_B^2 \mathbf{Q}_B^* \mathbf{w}_B}} \\
&= \frac{\mathbf{v}_A^* \tilde{\mathbf{C}}_{AB} \mathbf{v}_B}{\sqrt{\mathbf{v}_A^* \mathbf{v}_A \; \mathbf{v}_B^* \mathbf{v}_B}} \\
&= \hat{\mathbf{v}}_A^* \tilde{\mathbf{C}}_{AB} \hat{\mathbf{v}}_B
\end{aligned} \tag{7.9}$$

where $\hat{\mathbf{v}}_A$ and $\hat{\mathbf{v}}_B$ denote normalized unit vectors of $\mathbf{v}_A$ and $\mathbf{v}_B$. This expression of ρ is simple to maximize with respect to $\hat{\mathbf{v}}_A$ and $\hat{\mathbf{v}}_B$. At first thought, one might worry about what happens in the singular case, where eq. 7.7 impose the constraint that some elements of $\mathbf{v}_A$ and $\mathbf{v}_B$ have to be zero. These constraints are automatically satisfied at the maximum of eq. (7.9) since the forbidden subspaces are the same as the left and right nullspaces of $\tilde{\mathbf{C}}_{AB}$. To find the maximum, singular value decomposition (SVD) is applied on $\tilde{\mathbf{C}}_{AB}$.

$$\begin{aligned}
\tilde{\mathbf{C}}_{AB} &= \begin{pmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 & \dots \end{pmatrix} \begin{pmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \sigma_3 & \\ & & & \ddots \end{pmatrix} \begin{pmatrix} \mathbf{f}_1^* \\ \mathbf{f}_2^* \\ \mathbf{f}_3^* \\ \vdots \end{pmatrix} \\
&= \sum_k \sigma_k \mathbf{e}_k \mathbf{f}_k^*
\end{aligned} \tag{7.10}$$

By convention, $\{\mathbf{e}_i\}$ and $\{\mathbf{f}_i\}$ are both sets of orthonormal vectors. The singular values are real and sorted in descending order, i.e. $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq ... \geq 0$. The

[4] The pseudo inverse of a diagonal matrix is simple. Just invert each of the nonzero elements. For example $\begin{pmatrix} 2 & 0 \\ 0 & 0 \end{pmatrix}^\dagger = \begin{pmatrix} 0.5 & 0 \\ 0 & 0 \end{pmatrix}$

maximum is obtained when

$$\begin{aligned} \hat{\mathbf{v}}_A &= \mathbf{e}_1 \\ \hat{\mathbf{v}}_B &= \mathbf{f}_1 \\ \rho &= \sigma_1 \end{aligned} \tag{7.11}$$

Note that the SVD is not uniquely defined in case two or more singular values are equal. If the multiplicity of the largest singular value is greater than 1, the optimal $\mathbf{v}_A$ and $\mathbf{v}_B$ are not unique.

Finally, $\mathbf{w}_A$ and $\mathbf{w}_B$ can be solved, using eq. (7.7). This solution is ambiguous in the singular case, but pseudo inverse yields the smallest $\mathbf{w}_A$ and $\mathbf{w}_B$.

$$\mathbf{w}_A = \mathbf{Q}_A \mathbf{D}_A^\dagger \mathbf{v}_A \quad \text{and} \quad \mathbf{w}_B = \mathbf{Q}_B \mathbf{D}_B^\dagger \mathbf{v}_B \tag{7.12}$$

7.5 Canonical Variates

We do not know any good definition of what a canonical variate is. Borga[5] provides a definition that depends on his maximization method. In this thesis, a different maximization method is used and a different definition need to be introduced. In section 7.6 this definition is proved to be equivalent with Borga's definition.

In this thesis, canonical variates are defined as the (suboptimal) solutions to the canonical correlation corresponding to the different singular values in the SVD, eq. (7.10). The variate of index k is what we get if we replace eq. (7.11) with

$$\begin{aligned} \hat{\mathbf{v}}_A &= \mathbf{e}_k \\ \hat{\mathbf{v}}_B &= \mathbf{f}_k \\ \rho &= \sigma_k \end{aligned} \tag{7.13}$$

7.6 Equivalence with Borga's Solution

The objective of this section is to show that the CCA-SVD method gives the same solutions as Borga's[5] method that transforms the maximization problem to a generalized eigenvector problem. The following equations are valid, even for complex and singular cases. Thus, the equivalence proof also confirms the validity of Borga's[5] method for complex variables. Even the canonical variates are the same as in Borga's method.

Remember the singular value decomposition in eq.(7.10), $\tilde{\mathbf{C}}_{AB} = \sum_k \sigma_k \mathbf{e}_k \mathbf{f}_k^*$ and study what it means for the solutions of the following equation system.

$$\begin{aligned} \tilde{\mathbf{C}}_{AB} \hat{\mathbf{v}}_B &= \rho \hat{\mathbf{v}}_A \\ \tilde{\mathbf{C}}_{AB}^* \hat{\mathbf{v}}_A &= \rho \hat{\mathbf{v}}_B \end{aligned} \tag{7.14}$$

These equations are satisfied if and only if $\mathbf{v}_A$, $\mathbf{v}_B$ and ρ are the corresponding components in SVD of $\tilde{\mathbf{C}}_{AB}$. Or to be exact, in case the singular value has multiplicity ≥ 1, the solutions are linear combinations of SVD-vectors with the same

singular value. For readability, the linear combinations are not written explicitly, but they are implicit since the singular value decomposition is not unique.

$$\begin{aligned} \hat{\mathbf{v}}_A &= \mathbf{e}_k \\ \hat{\mathbf{v}}_B &= \mathbf{f}_k \qquad k = 1, 2, 3, \ldots \\ \rho &= \sigma_k \end{aligned} \tag{7.15}$$

This means that the canonical variates computed by our novel SVD method are the only solutions to eq. (7.14). We want these equations in the **w**-coordinates, as in Borga's thesis. Use eq. (7.7) to substitute $\mathbf{v}_A$ and $\mathbf{v}_B$. We also multiply the whole equations with $\mathbf{Q}_A\mathbf{D}_A$ and $\mathbf{Q}_B\mathbf{D}_B$ respectively. Despite we multiply with $\mathbf{D}_A$ that might be singular, we have equivalence with eq. (7.14), (since $\mathbf{D}_A$ and $\mathbf{D}_A^2$ have the same rank).

$$\mathbf{Q}_A\mathbf{D}_A\ \mathbf{D}_A^\dagger\mathbf{Q}_A^*\mathbf{C}_{AB}\mathbf{Q}_B\mathbf{D}_B^\dagger\ \mathbf{D}_B\mathbf{Q}_B^*\mathbf{w}_B = \mathbf{Q}_A\mathbf{D}_A\ \rho\mathbf{D}_A\mathbf{Q}_A^*\mathbf{w}_A \tag{7.16}$$

$$\mathbf{Q}_B\mathbf{D}_B\ \mathbf{D}_B^\dagger\mathbf{Q}_B^*\mathbf{C}_{AB}^*\mathbf{Q}_A\mathbf{D}_A^\dagger\ \mathbf{D}_A\mathbf{Q}_A^*\mathbf{w}_A = \mathbf{Q}_B\mathbf{D}_B\ \rho\mathbf{D}_B\mathbf{Q}_B^*\mathbf{w}_B \tag{7.17}$$

Thanks to eq. (A.2) makes most of the matrix product cancel out. We arrive at the following expression which is equivalent to eq.(4.30) in Borga's thesis, except for that the **w** vectors are not normalized.

$$\mathbf{C}_{AB}\mathbf{w}_B = \rho\mathbf{C}_{AA}\mathbf{w}_A \tag{7.18}$$

$$\mathbf{C}_{AB}^*\mathbf{w}_A = \rho\mathbf{C}_{BB}\mathbf{w}_B \tag{7.19}$$

We can normalize the vectors, provided we multiply rhs in one eq by "$\frac{\mu_A}{\mu_B}$" and the other rhs by "$(\frac{\mu_A}{\mu_B})^{-1}$" (Borga's variable names). Then we have exactly equation (4.30) in Borga's PhD thesis[5]. The singular values and vectors correspond to the canonical variates. Note that this proof holds even for the complex and singular cases.

To emphasize this is the generalized eigenvalue problem, let's write it in matrix form

$$\left(\begin{pmatrix} 0 & \mathbf{C}_{AB} \\ \mathbf{C}_{AB}^* & 0 \end{pmatrix} - \rho \begin{pmatrix} \mathbf{C}_{AA} & 0 \\ 0 & \mathbf{C}_{BB} \end{pmatrix} \right) \begin{pmatrix} \mathbf{w}_A \\ \mathbf{w}_B \end{pmatrix} = \mathbf{0} \tag{7.20}$$

We do not recommended Borga's method when $\mathbf{C}_{AA}$ and $\mathbf{C}_{BB}$ are close to singular. Experimental results[5] indicate serious numerical problems.

[5] our motion algorithms using Matlab function eig()

Chapter 8

Motion Estimation using Canonical Correlation

Canonical correlation has been successfully used for estimation of disparity in a stereo algorithm by Borga[5]. An important advantage of that method is the ability to handle depth discontinuities. Whereas conventional stereo algorithms smoothen disparity estimates across discontinuities, Borga's algorithm responds with a distinct discontinuity. Experiments with transparent layers even prove an ability to estimate multiple disparities at a single point in the image.

It should be pointed out that there are other stereo algorithms that can handle depth discontinuities, e.g. Birchfield-Tomasi[4] that searches for single pixel correspondence.

One may wish there were a motion estimation algorithm with the same advantages as Borga's stereo algorithm. In case of occlusion, one wish motion discontinuities would be be correctly estimated. One may also wish that transparent layers would give multiple motion estimates at a single point. Unfortunately, it is more complicated in motion estimation, due to the generalized aperture problem, described in chapter 5. It may still be possible to compute motion constraints are not smoothed across discontinuities and not much degraded by interference of multiple layers.

We have extended the stereo algorithm to estimate motions, but so far for only one motion. It remains to explore its potential abilities in estimation of multiple motions.

8.1 Operations Applied Locally in the Image.

The image is first convolved with a number of quadrature filters and then divided into patches, e.g. blocks of size 16x16 pixels. The patch should be so small that the motion can be considered as pure translation within the patch. Each of these patches for each of these filter outputs are processed independently to get a motion constraint, $\mathbf{c}$. This section describes these local operations.

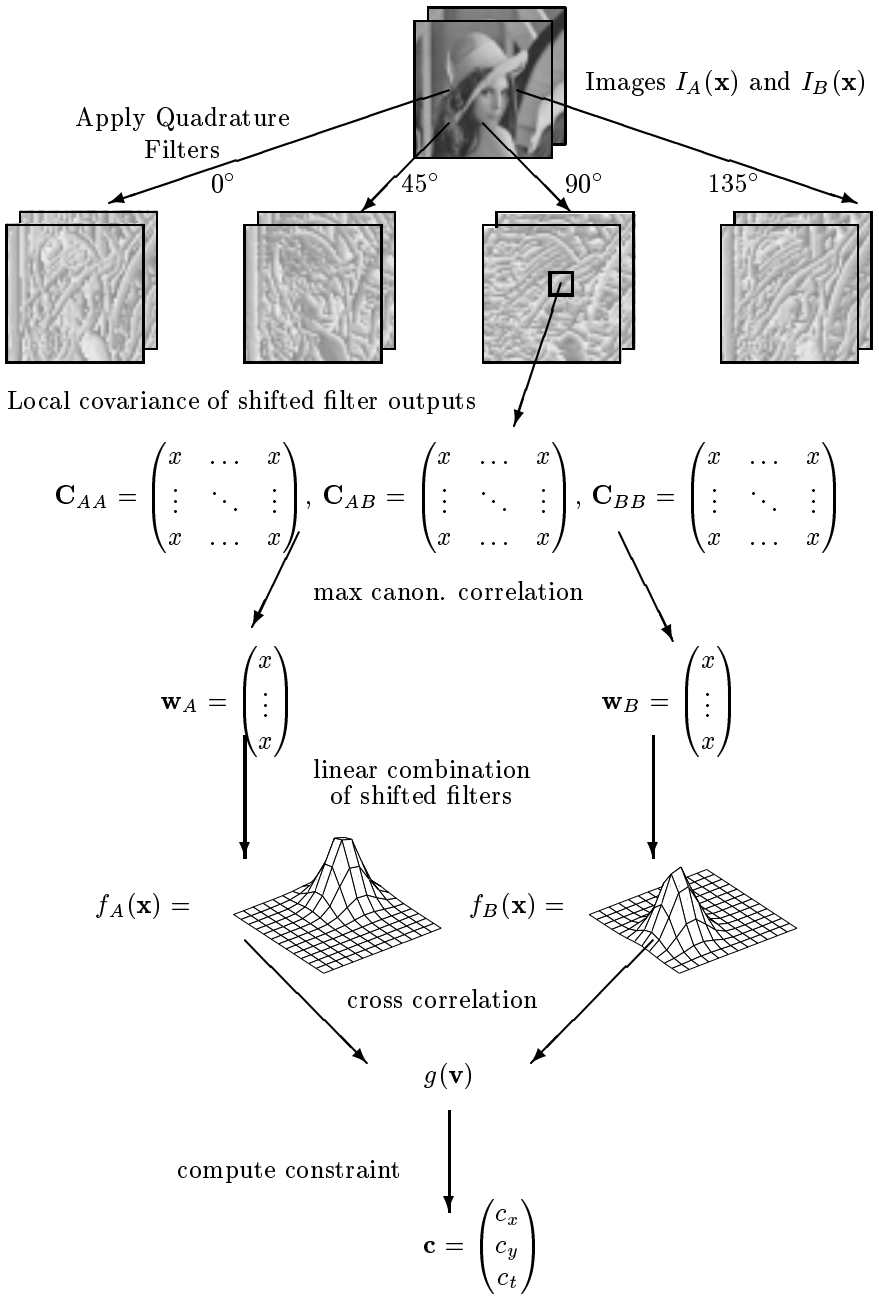

Figure 8.1: *From image to motion constraint for one direction and one patch. Don't forget that all values in between are complex numbers. A look up table can speed up computations.*

8.1.1 Shifted Quadrature Filter Outputs

Each of the two original images are convolved with a number of quadrature filters, as defined in section 1.4. We have used filters in directions 0, 45, 90 and 135 degrees. Since only one filter is used to compute one motion constraint, $\mathbf{c}$, the direction is dropped in our notations. For readability, we let $f(\mathbf{x})$ denote the quadrature filter of any directions. We have not tried filters with different center frequencies, but we believe it would improve performance.

$$q_A(\mathbf{x}) = (f * I_A)(\mathbf{x}) \quad \text{and} \quad q_B(\mathbf{x}) = (f * I_B)(\mathbf{x}) \tag{8.1}$$

These filter outputs are shifted with a number of predefined shifts, $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \ldots$ and correlated.

For example, in case the motion is exactly $\mathbf{v} = \mathbf{s}_3$ then $q_A(\mathbf{x})$ and $q_B(\mathbf{x} + \mathbf{s}_3)$ will make perfect correlation. In case we would have $\mathbf{v} \approx \mathbf{s}_3$ then we also get a high magnitude of correlation, but the value is complex with an argument almost proportional to the difference $\mathbf{v} - \mathbf{s}_3$. This property is fundamental in the phase-based method in chapter 4.

The method in this chapter is based on finding linear combinations of shifted filter outputs

$$\sum_i w_{Ai}\, q_A(\mathbf{x} + \mathbf{s}_i) \quad \text{and} \quad \sum_i w_{Bi}\, q_B(\mathbf{x} + \mathbf{s}_i). \tag{8.2}$$

that have highest possible correlation. The coefficients are complex and we arrange them in vectors

$$\mathbf{w}_A = \begin{pmatrix} w_{A1} \\ w_{A2} \\ w_{A3} \\ \vdots \end{pmatrix} \quad \text{and} \quad \mathbf{w}_B = \begin{pmatrix} w_{B1} \\ w_{B2} \\ w_{B3} \\ \vdots \end{pmatrix}. \tag{8.3}$$

8.1.2 Canonical Correlation

For each filter direction and each patch, canonical correlation is used to find the linear combinations of shifted filter outputs in eq (8.2) that have maximum correlation. The patch region in the image is denoted $\mathcal{N}$. The unknown coefficients in the linear combinations are organized in vectors $\mathbf{w}_A$ and $\mathbf{w}_B$. In terms of these notations, we want to maximize the following correlation under constraint it is real and positive.

$$\begin{aligned} \rho &= \max_{\mathbf{w}_A, \mathbf{w}_B} \frac{\iint_{\mathcal{N}} (\sum_i w_{Ai} q_A(\mathbf{x} + \mathbf{s}_i))^* \sum_i w_{Bi} q_B(\mathbf{x} + \mathbf{s}_i)\, d\mathbf{x}}{\sqrt{\iint_{\mathcal{N}} |\sum_i w_{Ai} q_A(\mathbf{x} + \mathbf{s}_i)|^2\, d\mathbf{x} \iint_{\mathcal{N}} |\sum_i w_{Bi} q_B(\mathbf{x} + \mathbf{s}_i)|^2\, d\mathbf{x}}} \\ &= \max_{\mathbf{w}_A, \mathbf{w}_B} \frac{\mathbf{w}_A^* \mathbf{C}_{AB} \mathbf{w}_B}{\sqrt{\mathbf{w}_A^* \mathbf{C}_{AA} \mathbf{w}_A\, \mathbf{w}_B^* \mathbf{C}_{BB} \mathbf{w}_B}} \end{aligned} \tag{8.4}$$

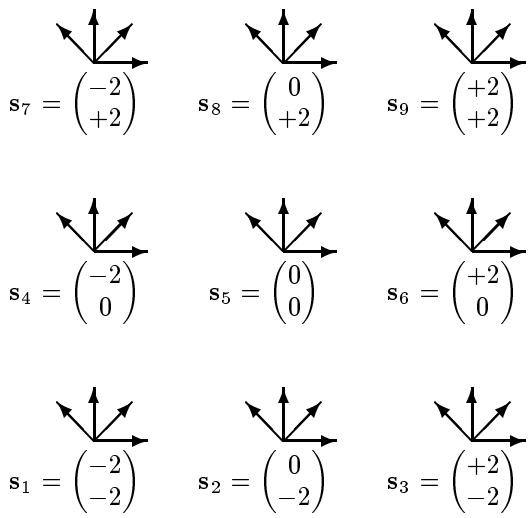

Figure 8.2: *A set of shifted quadrature filters in directions 0, 45, 90, 135 degrees that are used in experiments in section 8.4.*

This is the form of canonical correlation where $\mathbf{C}_{AA}$, $\mathbf{C}_{AB}$ and $\mathbf{C}_{BB}$ are covariance matrices. The element at row m and column n in each covariance matrix is computed as

$$C_{AA,mn} = \iint_{\mathcal{N}} q_A(\mathbf{x} + \mathbf{s}_m)^* \, q_A(\mathbf{x} + \mathbf{s}_n) \, d\mathbf{x} \tag{8.5}$$

$$C_{AB,mn} = \iint_{\mathcal{N}} q_A(\mathbf{x} + \mathbf{s}_m)^* \, q_B(\mathbf{x} + \mathbf{s}_n) \, d\mathbf{x} \tag{8.6}$$

$$C_{BB,mn} = \iint_{\mathcal{N}} q_B(\mathbf{x} + \mathbf{s}_m)^* \, q_B(\mathbf{x} + \mathbf{s}_n) \, d\mathbf{x} \tag{8.7}$$

The canonical correlation is maximized using the SVD-based method in chapter 7 that can handle singular covariance matrices. Matrices are virtually never singular, just ill conditioned, and therefore we threshold eigenvalues of the covariance matrices in eq. (7.6). The threshold should be much higher than what can be justified by errors in floating point arithmetics. In order to reject weak features in the images, the threshold in our implementation is set to $1/1000$ of the largest eigenvalue. The exact value of the threshold is probably not important and can vary several orders of magnitude without significant changes in motion estimates.

8.1.3 Correlation of Filters

Maximization of canonical correlation means finding the linear combinations of filter outputs that yield maximum correlation. In previous sections, we found the vectors of coefficients, $\mathbf{w}_A$ and $\mathbf{w}_B$, such that the maximimum correlation is obtained for

$$\sum_i w_{Ai}\, I_A * f(\mathbf{x}+\mathbf{s}_i) \quad \text{and} \quad \sum_i w_{Bi}\, I_B * f(\mathbf{x}+\mathbf{s}_i). \tag{8.8}$$

Thanks to the properties of convolution, it makes sense to study the linear combination of filters

$$f_A(\mathbf{x}) = \sum_i w_{Ai}\, f(\mathbf{x}+\mathbf{s}_i) \quad \text{and} \quad f_B(\mathbf{x}) = \sum_i w_{Bi}\, f(\mathbf{x}+\mathbf{s}_i) \tag{8.9}$$

instead of the filter outputs. Convolving the images with these filters is the same as convolving the image with each of the original filters and then computing linear combinations. In sense of correlation, these are the best possible linear combination of original filters. The motion can be estimated by analyzing these filters.

The filters obtained by linear combinations of quadrature filters in the same direction, are also quadrature filters. This statement is obvious if we think of the filter summation in the Fourier domain. Since all the added filters are zero in one half plane, the sum is also zero in one half plane.

Since the two images are similar except for a shift, i.e. $I_A(\mathbf{x}) = I_B(\mathbf{x}+\mathbf{v})$ the computed filters should also be similar, except for an equally large shift in opposite direction, $f_A(\mathbf{x}) = f_B(\mathbf{x}-\mathbf{v})$. To find the correct motion, $\mathbf{v}$, we analyze the cross correlation of the generated filters,

$$g(\mathbf{v}) = \iint f_A^*(\mathbf{x}+\mathbf{v})\, f_B(\mathbf{x})\, d\mathbf{x} \tag{8.10}$$

In a perfect world, the cross correlation of $g(\mathbf{v})$ has a peak value where $\mathbf{v}$ is the image motion. This peak value value is real and positive, i.e the phase crosses zero. In practice, the zero crossing of the phase does not perfectly coincide with the maximum amplitude. Just finding correlation peaks is of limited use in image regions that only have structure in one orientation, e.g. a straight line or edge. Phase is used since it is aware of the aperture problem. We also believe that zero crossings are more accurate than maximum amplitude. The phase of $g(\mathbf{v})$ crosses zero along curves in (v_x, v_y)-space. Usually, there are several curves, but the curve with the highest amplitude is probably the one corresponding to the image motion. How to analyze the cross correlation, $g(\mathbf{v})$, is described in section 8.1.5. The next section describes how to compute $g(\mathbf{v})$ using a look up table.

8.1.4 Look Up Table (LUT)

Since the generated filters, $f_A(\mathbf{x})$ and $f_B(\mathbf{x})$ are linear combinations of a set of original filters, their cross correlation, $g(\mathbf{v})$, is a sum of cross correlations of the original shifted filters. The LUT is computed by explicitly shifting filters. For

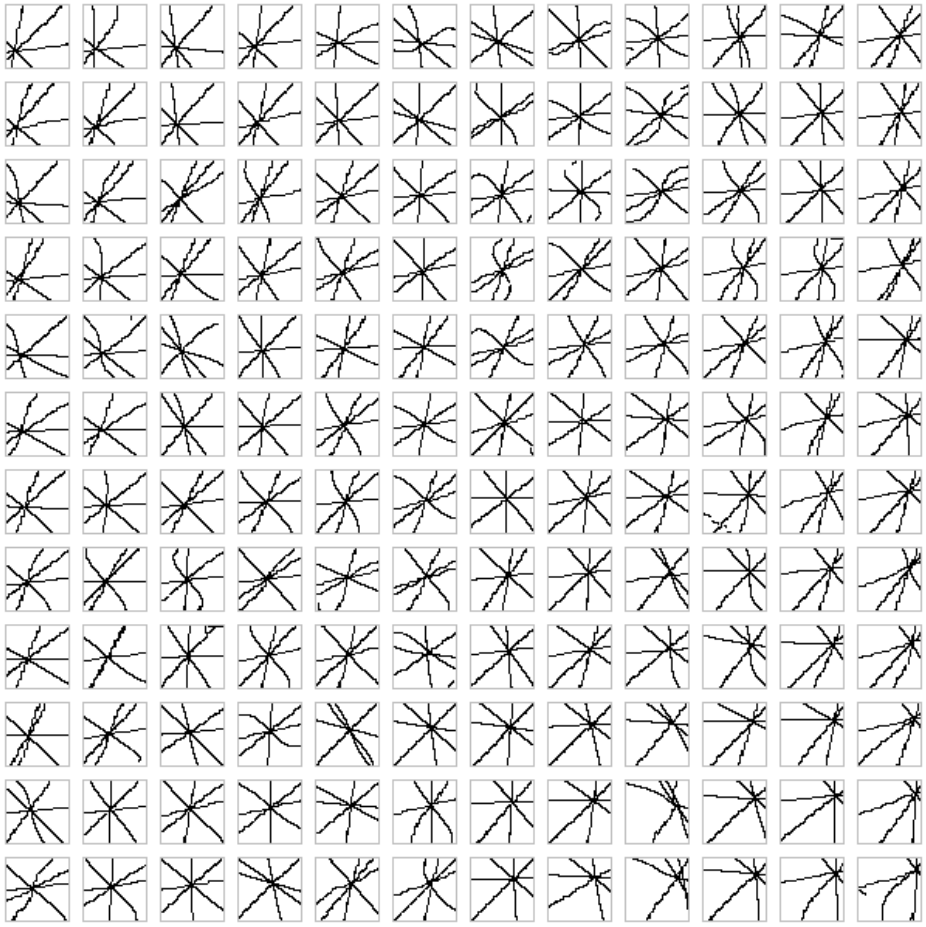

Figure 8.3: *Zero crossings of the phase for all the patches in an image with affine motion. In each subplot, the zero crossings,* $\arg g(\mathbf{v}) = 0$*, are drawn for each of the filter directions. Most zero crossings are straight lines. Sometimes, there are multiple false zero crossings. Since the motion is not pure translation, the intersections have different positions for different patches.*

subpixel accuracy, the shifts are implemented as multiplication in the Fourier domain. In matrix form the generated filters can be expressed as

$$\begin{aligned} f_A(\mathbf{x}) &= \sum_i w_{Ai} f(\mathbf{x} + \mathbf{s}_i) \\ &= \begin{pmatrix} f(\mathbf{x} + \mathbf{s}_1) & \dots & f(\mathbf{x} + \mathbf{s}_N) \end{pmatrix} \mathbf{w}_A. \end{aligned} \tag{8.11}$$

The cross correlation is a product of coefficients vectors from the canonical correlation and a matrix whose elements are cross correlation of the original filters.

$$\begin{aligned} g(\mathbf{v}) &= \iint f_A^*(\mathbf{x})\, f_B(\mathbf{x}+\mathbf{v})\, d\mathbf{x} \\ &= \iint \mathbf{w}_A^* \begin{pmatrix} f^*(\mathbf{x}+\mathbf{v}+\mathbf{s}_1) \\ f^*(\mathbf{x}+\mathbf{v}+\mathbf{s}_2) \\ \vdots \\ f^*(\mathbf{x}+\mathbf{v}+\mathbf{s}_N) \end{pmatrix} \begin{pmatrix} f(\mathbf{x}+\mathbf{s}_1) & \dots & f(\mathbf{x}+\mathbf{s}_N) \end{pmatrix} \mathbf{w}_B\, d\mathbf{x} \\ &= \mathbf{w}_A^* \iint \begin{pmatrix} f^*(\mathbf{x}+\mathbf{v}+\mathbf{s}_1) \\ f^*(\mathbf{x}+\mathbf{v}+\mathbf{s}_2) \\ \vdots \\ f^*(\mathbf{x}+\mathbf{v}+\mathbf{s}_N) \end{pmatrix} \begin{pmatrix} f(\mathbf{x}+\mathbf{s}_1) & \dots & f(\mathbf{x}+\mathbf{s}_N) \end{pmatrix} d\mathbf{x}\, \mathbf{w}_B \\ &= \mathbf{w}_A^*\, G(\mathbf{v})\, \mathbf{w}_B \end{aligned} \tag{8.12}$$

where

$$\mathbf{G}(\mathbf{v}) = \iint \begin{pmatrix} f^*(\mathbf{x}+\mathbf{v}+\mathbf{s}_1) \\ f^*(\mathbf{x}+\mathbf{v}+\mathbf{s}_2) \\ \vdots \\ f_N^*(\mathbf{x}+\mathbf{v}+\mathbf{s}_N) \end{pmatrix} \begin{pmatrix} f(\mathbf{x}+\mathbf{s}_1) & \dots & f(\mathbf{x}+\mathbf{s}_N) \end{pmatrix} d\mathbf{x} \tag{8.13}$$

$\mathbf{G}(\mathbf{v})$ is a look up table(LUT), that is precomputed for a number of different values of $\mathbf{v}$. Since subpixel shifts are necessary, there is an issue which interpolation method to use. For this particular data, we have chosen phase shifts in the Fourier domain, since we are not worried about ringings in the spatial domain. In order to reduce the effects of circular shifts, zeros are padded on the borders of the filters before computing FFT. Zero padding is equivalent to more dense sampling in the Fourier domain. For computational efficiency, Plancherel's formula may used to compute cross correlation directly in Fourier domain in order to avoid inverse FFT.

$$\begin{aligned} \mathbf{G}_{mn}(\mathbf{v}) &= \iint f^*(\mathbf{x}+\mathbf{v}+\mathbf{s}_m)\, f(\mathbf{x}+\mathbf{s}_n)\, d\mathbf{x} \\ &= \frac{1}{2\pi} \iint (F(\mathbf{u})\, e^{i\mathbf{u}^T(\mathbf{v}+\mathbf{s}_m)})^*\, F(\mathbf{u})\, e^{i\mathbf{u}^T\mathbf{s}_n}\, d\mathbf{u} \\ &= \frac{1}{2\pi} \iint |F(\mathbf{u})|^2\, e^{i\mathbf{u}^T(\mathbf{s}_n-\mathbf{s}_m-\mathbf{v})}\, d\mathbf{u} \end{aligned} \tag{8.14}$$

where $F(\mathbf{u})$ is the Fourier transform of $f(\mathbf{x})$.

In the next section, interpolation is used to compute $g(\mathbf{v})$ for values of $\mathbf{v}$ that are not in the look up table. Bilinear interpolation is used, but not directly on the real and imaginary parts. Instead, interpolation is done in polar representation of the complex numbers. The reason is because phase is more linear than the real and imaginary parts. This interpolation enables us to compute derivatives of the phase.

8.1.5 Motion Constraints from Correlation Data

The motion to estimate, $\mathbf{v}$, is assumed to be along one of the zero crossings of the phase of the correlation map. This yields a nonlinear constraint on the local motion, $\arg g(\mathbf{v}) = 0$.

In order to make computations reasonably simple, the nonlinear constraint is approximated by a linear motion constraint, $\mathbf{c}^T\overline{\mathbf{v}} = 0$, as defined in section 2.2, where notations are as in previous chapters,

$$\overline{\mathbf{v}} = \begin{pmatrix} \mathbf{v} \\ 1 \end{pmatrix} \quad \text{and} \quad \mathbf{c} = \begin{pmatrix} c_x \\ c_y \\ c_t \end{pmatrix}. \tag{8.15}$$

Unfortunately, there are often multiple zero crossings of the phase. In addition, the zero crossings are not along straight lines. For that reason, it is necessary to know roughly what the motion is before converting to a linear motion constraint. Assuming the motion is close to $\mathbf{v}_0$, we think that a linear motion constraint, $\mathbf{c}$, should have the following property

$$C \arg g(\mathbf{v}) = \mathbf{c}^T\overline{\mathbf{v}} + \mathcal{O}(\|\mathbf{v} - \mathbf{v}_0\|^2). \tag{8.16}$$

The solution is

$$\begin{pmatrix} c_x \\ c_y \end{pmatrix} = C \nabla \arg g(\mathbf{v}_0) \quad \text{and} \quad c_t = C \arg g(\mathbf{v}_0) - c_x v_{0,x} - c_y v_{0,y} \tag{8.17}$$

where C is a confidence measure set to

$$C = \begin{cases} (\frac{1}{\mu_1 - \rho} - \frac{1}{\mu_1 - \mu_2})^{\mu_3} \; |g(\mathbf{v})| \, \|\nabla \arg g(\mathbf{v})\| & \text{if } \rho > \mu_2 \\ 0 & \text{otherwise} \end{cases} \tag{8.18}$$

where $\mu_1 = 1.001$, $\mu_2 = 0.98$ and $\mu_3 = 1$ are constants chosen by studying a few experiments. Note that the magnitude of the gradient of the phase is included in both eq (8.17) and eq (8.18).

8.2 Fitting Motion Model to Data

The image is divided into patches that each yield as many motion constraints as there are directions of quadrature filters. These constraints are combined according to the theory of motion models in chapter 3 and produce a motion estimate. Instead of iterative refinement as described in chapter 4, we iterate without warping the image. Instead, the motion constraints are recomputed for updated motions as $\mathbf{v}_0$ in eq. (8.17) and figure 8.4.

8.3 Choosing Patch Size

A small patch often contains too little information. For the canonical correlation to have meaning, we must have at least as many pixels in a patch as many as the

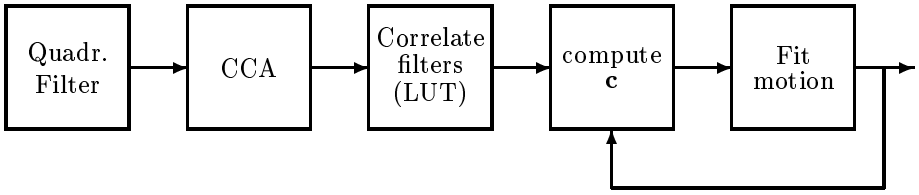

Figure 8.4: *Flow chart of our CCA-based motion estimation regarded from a single patch. Computation of motion constraints requires that the motion is known approximately. Since we can only make a rough guess, a number of iterations is necessary.*

number of shifts per filter. But even if there are fewer pixels, there is still a chance that canonical correlation finds good pair of linear combinations.

A too large patch, on the other hand, will not reflect the local structure in the image. It will rather tend to reflect the global distribution. Large patches also have problem to estimate motions that are not pure translations. Probably, the error in estimation of rotations is proportional to the patch size (for large patches). A few experiments on affine motions, suggest that the error is roughly a certain fraction of the variations within a single patch.

In addition, the larger patches are, the fewer they get and thus a lot of information seems to be thrown away. More patches yield more motion constraints.

8.4 Experimental Results

Figure 8.5 shows the accuracy in motion estimation on an image with synthetically introduced motions. The famous test image Lena512x512 has been shifted in several different directions and distances. Then the images have been subsampled to 128 pixels in order to hide the artifacts introduced by subpixel shifts. Thus, we have good test images of size 128x128 pixels and we know the answer. The motion is estimated and the mean square deviation is plotted for each magnitude of shifts. Since the look up table is only computed for shifts smaller than 2 pixels, it is impossible to estimate larger motions.

In the experiment, the center frequency of the filter is roughly 1 rad/pixel[1], the patch size is 16x16 pixels and the filter outputs are shifted according to $\mathbf{s}_i$ in figure 8.2.

[1]Filter is taken off the shelf with internal name is *orient8* in GOP.

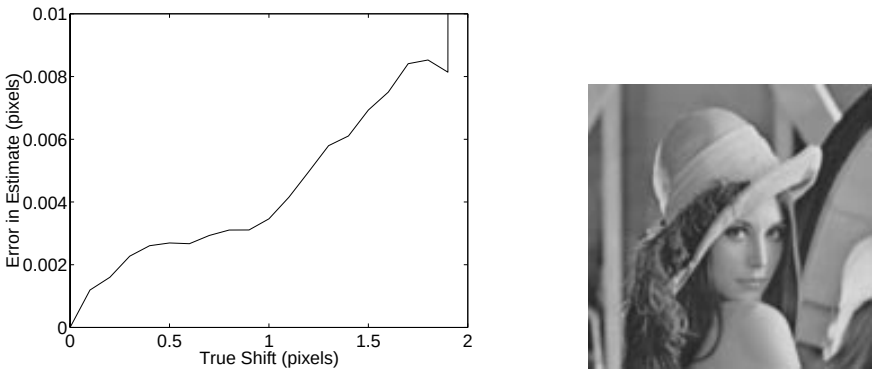

Figure 8.5: *Accuracy is very good for this synthetically shifted image, Lena128. The mean square error is plotted versus the amount the image is shifted. (Do not compare with figure 4.6 where no iterations are done.)*

8.5 Future Development

The experimental results on synthetic images is in itself a justification for the research we have done so far. Still our future goal is estimation of multiple motions, but which we still have not tried. The difficulties compared to the stereo algorithm is again the general aperture problem described in chapter 5.

8.5.1 Using Multiple Variates

The canonical correlation generates multiple canonical variates. Most of these canonical variates yield high correlation and similar cross correlation of generated filters $g(\mathbf{v})$. It may be possible to use more variates than the first one, but still we have not seen any significant improvement in experimental results.

8.5.2 Other Filters than Quadrature Filters

We have performed some experiments on replacing the quadrature filters by pair of odd and even real filters. The purpose is to allow more degrees of freedom in canonical correlation by allowing any linear combination of odd and even parts. Then $f_A(\mathbf{x})$ and $f_B(\mathbf{x})$ are sums of real filters that are both even and odd. Thus, the generated filters are not quadrature filters and the cross correlation must be done in a different way. In experiments, we have transform the filters to the Fourier domain and one of them is multiplied by $e^{i\phi}$ where ϕ denotes the angle in polar representation of the frequency in the Fourier domain. After cross correlation, then the magnitude of $g(\mathbf{v})$ is zero at the motion. Unfortunately, zero crossings of magnitude is harder to find than zero crossings of the phase. In particular, it gets harder to estimate large motions since there are more zero crossings of the magnitude.

8.5.3 Reducing Patch Size

Maybe reducing the patch size helps in estimation non-translational motions and motions of multiple transparent layers. It may be possible to reduce the patch size if fewer shifts, $\mathbf{s}_i$, are used. Maybe, the set of shifts should depend on direction of the filter. It may also help to have different shifts for the two images in case the motion is roughly known, as in iterative refinement. In the extreme case, if only one shift is used for every image, we get something similar to the phase-based method in chapter 4 with warp.

We suggest to be careful with such approaches. For example, only choosing shifts along a line would mean that the motion is estimated in a separable fashion and we are back at the mistake described in section 2.1.1.

Appendices

A Details for Chapter 7 on Canonical Correlation

A.1 Failure to Compute Derivative with Respect to a Complex Variable

Calculus with complex variables often obey the same rules as with real variables. For that reason, it is easy to forget that the same rules do not always apply. As interest for this chapter, we will show why it is not possible to compute the derivative of a complex conjugate. Let $f(z) = z^*$ and the derivative is defined as a limit that does not exist since h is a complex number,

$$\begin{aligned} f'(z) &= \lim_{|h|\to 0} \frac{f(z+h) - f(z)}{h} \\ &= \lim_{|h|\to 0} \frac{(z+h)^* - \overline{z}}{h} \\ &= \lim_{|h|\to 0} \frac{h^*}{h}. \end{aligned} \tag{A.1}$$

Of course, it is still possible to split the complex variable into real and imaginary parts, $z = a + ib$ and then calculate the partial derivatives $\frac{\partial f(a+ib)}{\partial a}$ and $\frac{\partial f(a+ib)}{\partial b}$.

A.2 Beginner's Example of Canonical Correlation

Assume X_1, X_2, X_3, X_4 are independent stochastic variables, with zero mean and standard deviation = 1.

$$\mathbf{z}_A = \begin{pmatrix} X_1 + X_2 \\ X_1 - X_2 \\ X_3 \end{pmatrix} \quad \text{and} \quad \mathbf{z}_B = \begin{pmatrix} X_1 \\ X_4 \end{pmatrix}$$

Note that the only variable that appears in both data set is X_1. For this simple case, it is obvious that maximum correlation is between $z_{A,1} + z_{A,2}$ and $z_{B,1}$.

To verify the CCA algorithm, go through the formal computations.

$$\mathbf{C}_{AA} = E[\mathbf{z}_A^{T*}\mathbf{z}_A^T] = E[\begin{pmatrix} (X_1+X_2)^*(X_1+X_2) & (X_1+X_2)^*(X_1-X_2) & (X_1+X_2)^*X_3 \\ (X_1-X_2)^*(X_1+X_2) & (X_1-X_2)^*(X_1-X_2) & (X_1-X_2)^*X_3 \\ X_3^*(X_1+X_2) & X_3^*(X_1-X_2) & X_3^*X_3 \end{pmatrix}] = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{C}_{AB} = E[\mathbf{z}_A^{T*}\mathbf{z}_B^T] = E[\begin{pmatrix} (X_1+X_2)^*X_1 & (X_1+X_2)^*X_4 \\ (X_1-X_2)^*X_1 & (X_1-X_2)^*X_4 \\ X_3^*X_1 & X_3^*X_4 \end{pmatrix}] = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\mathbf{C}_{BB} = E[\mathbf{z}_B^{T*}\mathbf{z}_B^T] = E[\begin{pmatrix} X_1^*X_1 & X_1^*X_4 \\ X_4^*X_1 & X_4^*X_4 \end{pmatrix}] = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Maximization gives

$$\mathbf{w}_A = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{w}_B = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad \rho = 1$$

This time, $\rho = 1$, which means that the two linear combinations are always equal, except for a scalar factor. This is not normal in real world application, where it is possible where no linear combination makes perfect correlation.

A.3 Proof of Equation (7.9)

This section is a proof that given the variable definitions in chapter 7

$$\mathbf{D}_A\mathbf{D}_A^\dagger\mathbf{Q}_A^*\mathbf{C}_{AB}\mathbf{Q}_B\mathbf{D}_B^\dagger\mathbf{D}_B = \mathbf{Q}_A^*\mathbf{C}_{AB}\mathbf{Q}_B \tag{A.2}$$

In the singular case, $\mathbf{D}_A\mathbf{D}_A^\dagger \neq \mathbf{I}$ and(or) $\mathbf{D}_B^\dagger\mathbf{D}_B \neq \mathbf{I}$, but we will show that eq (7.9) is still valid thanks to the relations between $\mathbf{C}_{AA}$, $\mathbf{C}_{AB}$ and $\mathbf{C}_{BB}$. It is enough to prove one half of the theorem

$$\mathbf{Q}_A^*\mathbf{C}_{AB} = \mathbf{D}_A\mathbf{D}_A^\dagger\mathbf{Q}_A^*\mathbf{C}_{AB} \tag{A.3}$$

The other part of the theorem, $\mathbf{C}_{AB}\mathbf{Q}_B = \mathbf{C}_{AB}\mathbf{Q}_B\mathbf{D}_B^\dagger\mathbf{D}_B$, can be proved in the same way.

Before going into the core of the proof, note that $\mathbf{D}_A\mathbf{D}_A^\dagger$ is equal to the identity matrix, except in the positions where $\mathbf{D}_A$ is zero[1]. The definitions of these covariance matrices implies that a null space in the $\mathbf{C}_{AA}$ and $\mathbf{C}_{BB}$ are also left

[1]For example: If $\mathbf{D}_A = \begin{pmatrix} 3.26 & 0 & 0 \\ 0 & 56.31 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ then $\mathbf{D}_A\mathbf{D}_A^\dagger = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

and right null spaces of $\mathbf{C}_{AB}$. Let's apply a coordinate transformation to obtain a form of the canonical correlation which is useful in the proof.

$$\begin{aligned}\rho &= \frac{\mathbf{w}_A^* \mathbf{Q}_A \mathbf{Q}_A^* \mathbf{C}_{AB} \mathbf{Q}_B \mathbf{Q}_B^* \mathbf{w}_B}{\sqrt{\mathbf{w}_A^* \mathbf{Q}_A \mathbf{D}_A^2 \mathbf{Q}_A^* \mathbf{w}_A \; \mathbf{w}_B^* \mathbf{Q}_B \mathbf{D}_B^2 \mathbf{Q}_B^* \mathbf{w}_B}} \\ &= \frac{\mathbf{u}_A^* \mathbf{Q}_A^* \mathbf{C}_{AB} \mathbf{Q}_B \mathbf{u}_B}{\sqrt{\mathbf{u}_A^* \mathbf{D}_A^2 \mathbf{u}_A \; \mathbf{u}_B^* \mathbf{D}_B^2 \mathbf{u}_B}}\end{aligned} \tag{A.4}$$

where

$$\begin{aligned}\mathbf{u}_A &= \mathbf{Q}_A^* \mathbf{w}_A \\ \mathbf{u}_B &= \mathbf{Q}_B^* \mathbf{w}_B\end{aligned} \tag{A.5}$$

Here comes the core of the proof. Let's pick arbitrary $\mathbf{u}_A$ and $\mathbf{u}_B$ and split the former into parts

$$\begin{aligned}\mathbf{u}_{A\|} &= \mathbf{D}_A \mathbf{D}_A^\dagger \, \mathbf{u}_A \\ \mathbf{u}_{A\perp} &= (\mathbf{I} - \mathbf{D}_A \mathbf{D}_A^\dagger) \, \mathbf{u}_A\end{aligned} \tag{A.6}$$

Note that $\mathbf{u}_{A\|} + \mathbf{u}_{A\perp} = \mathbf{u}_A$. We will prove that the error in the numinator of eq. (A.4) is zero.

$$\begin{aligned}\varepsilon &= \mathbf{u}_A^* \mathbf{Q}_A^* \mathbf{C}_{AB} \mathbf{Q}_B \mathbf{u}_B - \mathbf{u}_A^* \mathbf{D}_A \mathbf{D}_A^\dagger \mathbf{Q}_A^* \mathbf{C}_{AB} \mathbf{Q}_B \mathbf{u}_B \\ &= \mathbf{u}_A^* (\mathbf{I} - \mathbf{D}_A \mathbf{D}_A^\dagger) \mathbf{Q}_A^* \mathbf{C}_{AB} \mathbf{Q}_B \mathbf{u}_B \\ &= (\mathbf{u}_{A\|}^* + \mathbf{u}_{A\perp}^*)(\mathbf{I} - \mathbf{D}_A \mathbf{D}_A^\dagger) \mathbf{Q}_A^* \mathbf{C}_{AB} \mathbf{Q}_B \mathbf{u}_B \\ &= \mathbf{u}_{A\perp}^* \mathbf{Q}_A^* \mathbf{C}_{AB} \mathbf{Q}_B \mathbf{u}_B\end{aligned} \tag{A.7}$$

To prove that $\mathbf{u}_{A\perp}^* \mathbf{Q}_A^* \mathbf{C}_{AB}$ is zero, we employ a simple trick. Study the correlation, when the coefficients of the linear combinations are $\mathbf{u}_{A\perp}$ and $\mathbf{u}_B$.

$$\begin{aligned}\rho(\mathbf{u}_{A\perp}, \mathbf{u}_B) &= \frac{\mathbf{u}_{A\perp}^* \mathbf{Q}_A^* \mathbf{C}_{AB} \mathbf{Q}_B \mathbf{u}_B}{\sqrt{\mathbf{u}_{A\perp}^* \mathbf{D}_A^2 \mathbf{u}_{A\perp} \; \mathbf{u}_B^* \mathbf{D}_B^2 \mathbf{u}_B}} \\ &= \frac{\mathbf{u}_{A\perp}^* \mathbf{Q}_A^* \mathbf{C}_{AB} \mathbf{Q}_B \mathbf{u}_B}{\sqrt{0 \; \mathbf{u}_B^* \mathbf{D}_B^2 \mathbf{u}_B}} \\ &= \frac{\varepsilon}{0}\end{aligned} \tag{A.8}$$

Remind that the canonical correlation, $|\rho| \leq 1$, eq. (7.2.1). Thus, a zero in the denuminator means a zero in the numinator. Thus, $\varepsilon = 0$ for arbitrary $\mathbf{u}_A$ and $\mathbf{u}_B$, it follows that eq. (A.3) is proven.

B Variable Names

All variable names that are used without immediate explanation are listed here. Most variable names are local of each chapter, but there are also variables that are used throughout the thesis. The right column indicates where each variable is defined. An introduction to our style and notations (except for variable names) is provided in section 1.3.

B.1 Global Variable Names

The following notations are used in many chapters without immediate explanation at every occurence.

$\mathbf{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix}$	image motion	eq. (3.3)
$\mathbf{c} = \begin{pmatrix} c_x \\ c_y \\ c_t \end{pmatrix}$	motion constraint, such that $\mathbf{c}^T\mathbf{v} = 0$	eq. (2.2)
$\mathbf{x} = \begin{pmatrix} x \\ y \end{pmatrix}$	spatial position	eq. (3.2)
$\mathbf{x}_k$	often spatial coordinate of constraint, $\mathbf{c}_k$, with index k.	eq. (3.12)
$\mathbf{a}$	vector of parameters for motion model	eq. (3.7)
$\mathbf{K}(\mathbf{x})$	matrix of basis functions for motion model	eq. (3.8)
$I_A(\mathbf{x})$, $I_B(\mathbf{x})$	two images that are input to motion estimation	section 4.2.1

B.2 Local Variable Names in Chapter 3

$\overline{\mathbf{v}} = \begin{pmatrix} \mathbf{v} \\ 1 \end{pmatrix}$	image motion with an extra element=1	eq. (3.11)
$\overline{\mathbf{a}} = \begin{pmatrix} \mathbf{a} \\ 1 \end{pmatrix}$	model parameter vector with extra element=1	eq. (3.11)
$\overline{\mathbf{K}}(\mathbf{x})$	matrix of basis functions with extra element	eq. (3.11)
$\varepsilon(\mathbf{a})$	error when fitting model	eq. (3.12)
k	often index of motion constraint (joint index for spatial position, filter direction e.t.c.)	chapter 3
$\overline{\mathbf{Q}}$	Symmetric matrix defining quadratic form	eq. (3.13)
$\mathbf{Q}$, $\mathbf{q}$	submatrix/vector of $\overline{\mathbf{Q}}$	eq. (3.14)
$\mathbf{P}$	matrix defining cost function	section 3.3
λ	Scalar multiplier of cost	eq. (3.18)

B.3 Local Variable Names in Chapter 4

$f_j(\mathbf{x})$	Quadrature filter with index j	eq. (4.2)
$\hat{\mathbf{n}}$	Direction of quadrature filter.	eq. (4.1)
$q_{A,j}(\mathbf{x})$, $q_{B,j}(\mathbf{x})$	Output from quadrature filter with index j convolved with images A and B respectively	eq. (4.2)
$\theta_{A,j}(\mathbf{x})$	Phase computed from image A and filter j	eq. (4.3)
C	Confidence in constraint $\mathbf{c}$	eq. (4.4), eq. (4.10)

B.4 Local Variable Names in Chapter 5

M	number of layers
N	number of parameters in motion model

B.5 Local Variable Names in Chapter 6

This chapter uses the same notations as in chapter 3 and the following.

$\mathbf{a}_n$	parameters describing motion of layer n	
$m_{n,l}$	mixture probability - the probability of observing a constraint for layer n in a warped image with index l	
q_{nkl}	owner probability - the probability that the particular constraint $\mathbf{c}_{kl}$ belongs to layer n.	
n	often index of motion layer	
k	often index of motion constraint $\mathbf{c}_k$ (joint index for spatial position, filter direction e.t.c.)	
l	often the index of image warped according to estimated motion with index $(n =)j$	
$P(X\|Y)$	conditional probability density function of X when Y is known	
$\nabla_{\mathbf{a}}$	Gradient with respect to variables in vector $\mathbf{a}$	
$d(\mathbf{c}, \mathbf{v})$	distance from a motion to a given constraint	eq. (6.19)
$\varepsilon(\mathbf{a}_1, \mathbf{a}_2)$	Error measure in alternative method	eq. (6.25)
T_4^{ijkl}, T_3^{ijk}, T_2^{ij}, T_1^i, T_0	tensors with 4, 3, 2, 1, 0 indices of fourth moments of motion constraints	eq. (6.30)

B.6 Local Variable Names in Chapter 7

$\mathbf{z}_A$, $\mathbf{z}_B$	vectors of input data (stochastic variables)	section 7.1
$\mathbf{w}_A$, $\mathbf{w}_B$	vectors with coefficients for linear combination of the elements in $\mathbf{z}_A$ and $\mathbf{z}_B$. The correlation is maximized with respect to $\mathbf{w}_A$ and $\mathbf{w}_B$.	eq. (7.1)
z_A, z_B	Linear combination of stochastic variables $z_A = \mathbf{w}_A^T \mathbf{z}_A$	eq. (7.1)
ρ	canonical correlation	eq. (7.2)
$\mathbf{C}_{AA}$, $\mathbf{C}_{AB}$, $\mathbf{C}_{BB}$	covariance matrices	eq. (7.3)
$\mathbf{D}_A$, $\mathbf{D}_B$	Diagonal matrix in eigenvalue decomposition of $\mathbf{C}_{AA}$ and $\mathbf{C}_{BB}$. All elements are real and nonnegative.	eq. (7.6)
$\mathbf{Q}_A$, $\mathbf{Q}_A$	Transformation matrix in eigenvalue decomposition of $\mathbf{C}_{AA}$ and $\mathbf{C}_{BB}$. Complex elements and unitary.	eq. (7.6)
$\mathbf{v}_A$, $\mathbf{v}_B$	Transformed vectors of $\mathbf{w}_A$ and $\mathbf{w}_B$.	eq. (7.7)
$\tilde{\mathbf{C}}_{AB}$	Transformed covariance matrix $\mathbf{C}_{AB}$	eq. (7.8)
$\hat{\mathbf{v}}_A$, $\hat{\mathbf{v}}_B$	normalized unit vectors of $\mathbf{v}_A$ and $\mathbf{v}_B$	eq. (7.9)
$\mathbf{D}^\dagger$	Pseudo inverse of matrix $\mathbf{D}$.	eq. (7.8)
σ_k, $\mathbf{e}_k$, $\mathbf{f}_k$	SVD of $\tilde{\mathbf{C}}_{AB}$	eq. (7.10)

B.7 Local Variable Names in Chapter 8

$f(\mathbf{x})$	Quadrature filter (one out of several in different directions)	section 8.1.1
$q_A(\mathbf{x})$, $q_B(\mathbf{x})$	Outputs from some quadrature filter $f(\mathbf{x})$ applied on images $I_A(\mathbf{x})$ and $I_B(\mathbf{x})$.	eq. (8.1)
$\mathbf{s}_1, \mathbf{s}_2, \ldots$	shift of quadrature filter outputs when computing covariance matrices	section 8.1.1
$\mathbf{w}_A$, $\mathbf{w}_B$	coefficients for linear combination	eq. (8.3)
$\mathcal{N}$	patch region in the image	section 8.1.2
ρ	canonical correlation	eq. (8.4)
$\mathbf{C}_{AA}$, $\mathbf{C}_{AB}$, $\mathbf{C}_{BB}$,	covariance matrices for canonical correlation as in chapter 7	section 8.1.2
$f_A(\mathbf{x})$, $f_B(\mathbf{x})$	Generated filters. Linear combination of shifted original filters $f(\mathbf{x})$	eq. (8.9)
$g(\mathbf{v})$	cross correlation of generated filters – complex value	eq. (8.10)
$\mathbf{G}(\mathbf{v})$	Look up table (LUT) – a matrix for each $\mathbf{v}$.	eq. (8.13)

Bibliography

[1] V Torre A Verri, F Giroso. Differential techniques for optical flow. *Journal of the Optic Society of North America*, 7:912-922, 1990.

[2] M. Andersson and H. Knutsson. General sequential Spatiotemporal Filters for Efficient Low Level Vision. In *ECCV-96*, April 1996. Submitted.

[3] J. L. Barron, D. J. Fleet, S. S. Beauchemin, and T. A. Burkitt. Performance of optical flow techniques. In *Proc. of the CVPR*, pages 236–242, Champaign, Illinois, USA, 1992. IEEE. Revised report July 1993, TR-299, Dept. of Computer Science, University of Western Ontario, London, Ontario, Canada N6A 5B7.

[4] S. Birchfield and C. Tomasi. Depth discontinuities by pixel-to-pixel stereo.". *Proceedings of the IEEE International Conference on Computer Vision*, pages 1073–1080, January 1998.

[5] M. Borga. *Learning Multidimensional Signal Processing.* PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, 1998. Dissertation No 531, ISBN 91-7219-202-X.

[6] A. D. Calway, H. Knutsson, and R. Wilson. Multiresolution estimation of 2-d disparity using a frequency domain approach. In *Proc. British Machine Vision Conf.*, Leed, UK, September 1992.

[7] A. D. Calway, H. Knutsson, and R. Wilson. Multiresolution estimation of 2-d disparity using a frequency domain approach. In *Proc. British Machine Vision Conf.*, Leed, UK, September 1992.

[8] A. D. Calway, H. Knutsson, and R. Wilson. Multiresolution frequency domain algorithm for fast image registration. In *Proc. 3rd Int. Conf. on Visual Search*, Nottingham, UK, August 1992.

[9] G. Farnebäck. Motion-based Segmentation of Image Sequences. Master's Thesis LiTH-ISY-EX-1596, Computer Vision Laboratory, SE-581 83 Linköping, Sweden, May 1996.

[10] G. Farnebäck. Spatial Domain Methods for Orientation and Velocity Estimation. Lic. Thesis LiU-Tek-Lic-1999:13, Dept. EE, Linköping University,

SE-581 83 Linköping, Sweden, March 1999. Thesis No. 755, ISBN 91-7219-441-3.

[11] D. J. Fleet and A. D. Jepson. Computation of Component Image Velocity from Local Phase Information. *Int. Journal of Computer Vision*, 5(1):77–104, 1990.

[12] D. J. Fleet, A. D. Jepson, and M. R. M. Jenkin. Phase-based disparity measurement. *CVGIP Image Understanding*, 53(2):198–210, March 1991.

[13] G. H. Granlund and H. Knutsson. *Signal Processing for Computer Vision.* Kluwer Academic Publishers, 1995. ISBN 0-7923-9530-1.

[14] M. Hemmendorff, M. T. Andersson, and H. Knutsson. Phase-based image motion estimation and registration. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 1999*, Phoenix, AZ, USA, March 1999. IEEE.

[15] M. Hemmendorff, H. Knutsson, M. T. Andersson, and T. Kronander. Motion compensated digital subraction angiography. In *Proceedings of SPIE's International Symposium on Medical Imaging 1999*, volume 3661 Image Processing, San Diego, USA, February 1999. SPIE.

[16] Magnus Hemmendorff. Motion compensated digital subtraction angiography. Master's thesis, Linköpings universitet, 1997. LiTH-ISY-EX-1750.

[17] B. K. P. Horn and B. G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–204, 1981.

[18] M. Irani and S. Peleg. Motion analysis for image enhancement: resolution, occlusion, and transparency. *Journal of Visual Communications and Image Representation*, 4(4):324–335, 1993.

[19] A. Jepson and M. Black. Mixture models for optical flow. Technical Report RBCV-TR-93-44, Res. in Biol. and Comp. Vision, Dept. of Comp. Sci., Univ. of Toronto, 1993.

[20] A. D. Jepson and D. J. Fleet. Scale-space singularities. In O. Faugeras, editor, *Computer Vision-ECCV90*, pages 50–55. Springer-Verlag, 1990.

[21] J. Karlholm. *Local Signal Models for Image Sequence Analysis.* PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, 1998. Dissertation No 536, ISBN 91-7219-220-8.

[22] Scott R. Kerns and Jr. Irvin F.Hawkins. Carbon dioxide digital subtraction angiography. *AJR*, 164, 1995.

[23] H. Knutsson and M. Andersson. Optimization of Sequential Filters. In *Proceedings of the SSAB Symposium on Image Analysis*, pages 87–90, Linköping, Sweden, March 1995. SSAB. LiTH-ISY-R-1797. URL: http://www.isy.liu.se/cvl/ScOut/TechRep/TechRep.html.

[24] Centers for Disease Control Lewis A. Connor, David Satcher and Prevention. Reducing the burden of cardiovascular disease: Cdc strategies in evolution. Chronic Disease Notes and Reports, 1997.

[25] Jörg F. Debatin Martin R. Prince, Thomas M. Grist. *3D Contrast MR Angiography, 2nd edition*. Springer, 1999.

[26] G. J. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley, 1997.

[27] H. H. Nagel. On the estimation of optical flow: Relations between different approaches and som new results. *Artificial Intelligence*, 33:299–324, 1987.

[28] J. Whiting R. Close. Decomposition of coronary angiograms into non-rigid moving layers. In *Proceedings of SPIE's International Symposium on Medical Imaging 1999*, volume 3661 Image Processing, San Diego, USA, February 1999. SPIE.

[29] J. Shi and C. Tomasi. Good features to track. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[30] M. Shizawa and K. Mase. Simultaneous multiple optical flow estimation. In *Proceedings of the 10th International Conference on Pattern Recognition*, volume 1, pages 274–278, 1990.

[31] M. Shizawa and K. Mase. Principle of superposition: A common computational framework for analysis of multiple motion. In *IEEE Workshop on Visual Motion*, Princton, NJ, 1991.

[32] Jürgen Weese T. Buzug. Weighted least squares for point-based registration in digital subtraction angiography (dsa). In *Proceedings of SPIE's International Symposium on Medical Imaging 1999*, volume 3661 Image Processing, San Diego, USA, February 1999. SPIE.

[33] C-J. Westelius. *Focus of Attention and Gaze Control for Robot Vision*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, 1995. Dissertation No 379, ISBN 91-7871-530-X.

[34] C-F. Westin. *A Tensor Framework for Multidimensional Signal Processing*. PhD thesis, Linköping University, Sweden, SE-581 83 Linköping, Sweden, 1994. Dissertation No 348, ISBN 91-7871-421-4.

[35] Y. Zhu and N. J. Pelc. A spatiotemporal finite element mesh model of cyclical deforming motion and its application in myocardial motion analysis using phase contrast mr images. In *IEEE International Conference on Image Processing 97*, volume II, pages 117–120, Santa Barbara, October 1997. IEEE.